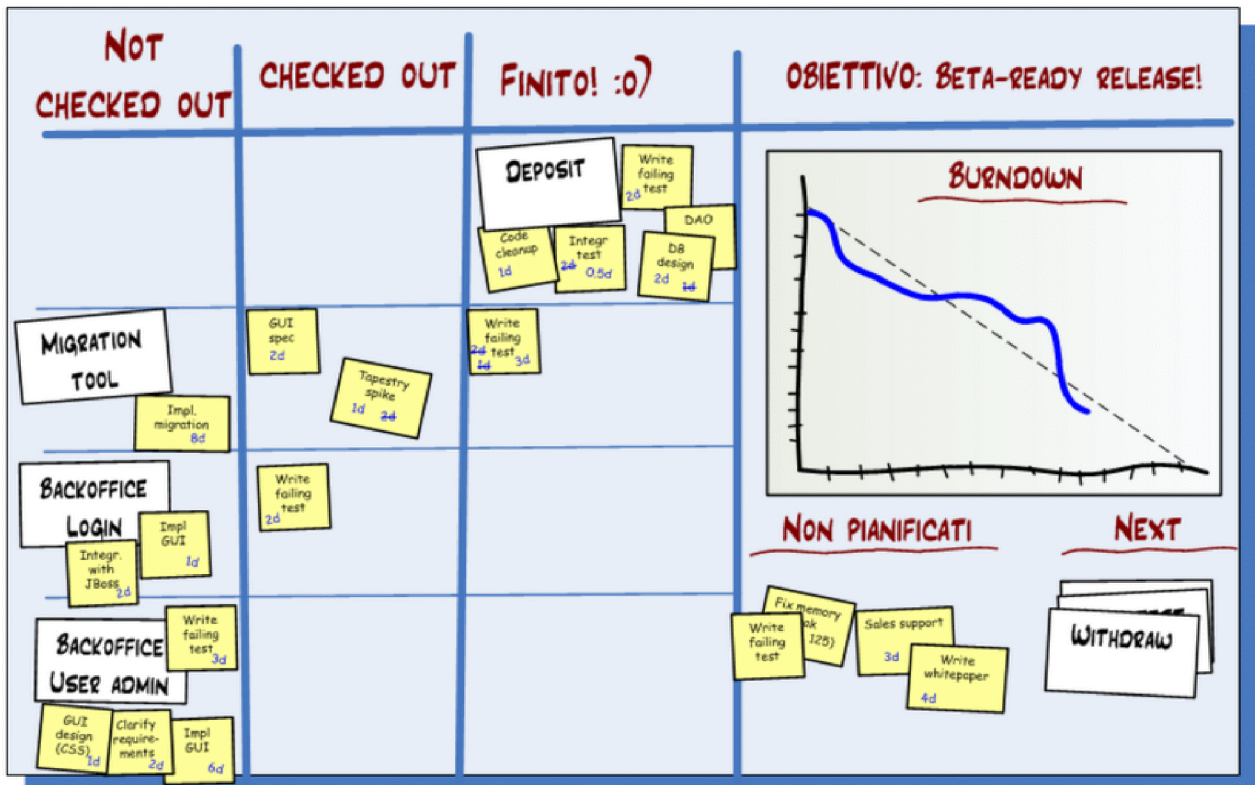


اسکرام و اکس پی

ساده شده

چگونه اسکرام را انجام می دهیم



Henrik Kniberg

تایید شده توسط مایک کان و جف سادرلند

مترجم و ویراستار:

Asad Safari

اسکرام و اکس پی ساده شده

چگونه اسکرام را انجام می دهیم

نویسنده : هنریک کنیبرگ

مترجم : اسد صفری

در این کتاب چه خواهید خواند

- چگونه بک لاگ محصول را ایجاد می کنیم
- چگونه برای برنامه ریزی اسپرینت آماده می شویم
- چگونه جلسه برنامه ریزی اسپرینت را انجام می دهیم
- چگونگی ارتباطات در طول اسپرینت
- چگونه بک لاگ اسپرینت را انجام می دهیم
- چگونه اتاق تیم را مرتب می کنیم
- چگونه جلسه روزانه اسکرام را انجام می دهیم
- چگونه دموی اسپرینت را ارائه می دهیم
- چگونه اسپرینت را بازبینی می کنیم
- چگونگی فاصله دهی مابین اسپرینت ها
- چگونگی انجام قرار دادهای قیمت بسته و برنامه ریزی Release
- چگونه XP را با اسکرام ترکیب می کنیم
- چگونه تست می کنیم
- چگونه چندین تیم اسکرام را اداره می کنیم
- چگونه تیم های پراکنده جغرافیایی را اداره می کنیم
- چک لیست مدیران اسکرام
- خواندنیهای مفید

سخن مترجم

بسیار خوشوقتم که توانستم اولین کتاب **Agile** و اسکرام را به زبان فارسی ترجمه و در اختیار علاقمندان این حوزه قرار دهم. به طور رسمی این اولین کتاب فارسی می باشد که به صورت کامل به موضوع اسکرام و مسائل مرتبط پرداخته است.

علت انتخاب این کتاب برای برگرداندن به فارسی روش های علمی و عملی جهت پیاده سازی و استفاده از اسکرام در داخل کتاب می باشد. نقطه فصل این کتاب با دیگر کتاب های اسکرام هم در همین جا می باشد که این کتاب بیشتر به جای توضیح مسائل تئوری روش های عملی را معرفی و شرح می دهد.

خوانندگان این کتاب می توانند به راحتی با داشتن حداقل آشنایی با اسکرام از این روش استفاده نمایند. علت اصلی ترجمه این کتاب به فارسی عدم وجود منبع و یا کتابی جامع در مورد اسکرام بود. من در طول این سال ها مقالات متعددی در مورد اسکرام و **Agile** به رشته تحریر در آورده بودم ولی با این حال همیشه جای خالی یک منبع کامل در مورد اسکرام را احساس می کردم. با توجه به اینکه حوزه کاری من در مورد آموزش و مربی گری اسکرام می باشد و بنابه تجربیات خودم در مورد اسکرام و مفید دیدن این کتاب برای صنعت نرم افزار ایران تصمیم به ترجمه این کتاب گرفتم.

در ترجمه انجام شده هیچ برداشت شخصی و یا نظر شخصی از طرف مترجم در متن اصلی وارد نشده است و تمام مطالب نظرات نویسنده کتاب می باشد. علاوه بر این هیچ قسمت و یا متنی از کتاب خلاصه سازی و یا حذف هم نشده است و دقیقاً ترجمه برابر اصل می باشد.

امیدوارم روش های معرفی شده در کتاب کارساز باشند و بتوانیم شاهد ایجاد تیم های متعدد اسکرام و **Agile** در سراسر ایران باشیم.

با تشکر

اسد صفری

Asad.safari@gmail.com

مربی و آموزگار **Agile**

مشاور سازمان ها و شرکت های توسعه نرم افزار در حوزه چابک سازی در **IrAgile**

مترجم بیانیه **Agile Manifesto**

برگزار کننده اولین دوره **CSM** در ایران

ابداع کننده بازی اسکرام گیم پک و اولین سنجش گر سازمان چابک

حقوق اثر و نحوه دسترسی به اثر

تمام حقوق مادی و معنوی این اثر برای مترجم محفوظ می باشد. نسخه پیش رو نسخه الکترونیکی رایگان از این کتاب می باشد و برای تهیه نسخه چاپی کتاب می توانید از روش های زیر استفاده کنید:

نحوه تهیه نسخه چاپی اثر

از طریق وب سایت : <http://www.irAgile.com>

از طریق وب لاگ دنیای چابک : <http://sirasad.wordpress.com>

اطلاعات تماس برای تهیه کتاب

ایمیل : Asad.safari@gmail.com

تقدیم

به جامعه Agile ایران

و

استاد مطلق Agile
Mike Cohn

و

خانواده مهربانم

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

1

مقدمه

شما به فکر استفاده از اسکرام در سازمان خود می باشید یا حتی از اسکرام چند ماه استفاده کرده اید . مبنای را یاد گرفته اید ، کتاب هایی را خوانده اید ، حتی شاید مدرک مدیر اسکرام¹ خود را نیز اخذ کرده اید . ولی هنوز احساس ناتوانی در این مورد می کنید.

در جملات کن شوئبر² (مبدع اسکرام) اسکرام یک متدلوژی نیست بلکه یک چارچوب یا فریم ورک می باشد . در واقع معنی جمله این می شود که اسکرام به شما نمی گوید دقیقا باید چه کاری انجام دهید .

خبر خوب این است که من می خواهم دقیقا آنگونه که اسکرام را انجام می دهیم به شما بگویم و خبر بد هم این است که این روش تنها یک روشی است که ما انجام می دهیم و این هم بدین معنی نیست که شما هم دقیقا عین این را انجام بدهید. در واقع شاید من در مواجه با وضعیت متفاوت ، اسکرام را به شیوه دیگری انجام دهم .

نقطه قوت و مشکل اسکرام این است که شما مجبورید اسکرام را با وضعیت خودتان و سازمانتان وفق دهید.

روش انجام اسکرام من نتیجه یک سال آزمایش اسکرام بر روی یک تیم توسعه 40 نفری می باشد. شرکتی در وضعیت سخت و بحرانی مانند اضافه کارهای زیاد ، مشکلات کیفیت حاد ، از دست دادن ضرب اعجل ها و این شرکت تصمیم گرفت تا از اسکرام استفاده نماید که پیاده سازی اسکرام

¹ Scrum Master

² Ken schwaber

وظیفه من بود. بیشتر افراد تیم توسعه اسم "اسکرام" را بدون اینکه حتی مفهوم واقعی آن را بدانند دائماً می شنیدند.

طی یک سال، ما اسکرام را در تمام لایه های شرکت پیاده سازی نمودیم. سعی کردیم از تعداد افراد متفاوت در تیم ها استفاده کنیم (3-12 نفر)، اندازه اسپرینت را تغییر دهیم (2-6 هفته)، تعریف "تکمیل شد" یا "Done" را تغییر دهیم، از فرمت های دیگری برای بک لاگ محصول استفاده بکنیم (Excel, Jira, Index Card)، از استراتژی های دیگر آزمایش کردن استفاده کنیم، به روش دیگری دموی محصول را ارائه کنیم، از روش های دیگر همسان سازی تیم های مختلف اسکرام استفاده کنیم و غیره. ما همچنین روش های XP را نیز مورد آزمایش قرار دادیم: سعی کردیم از روش های متفاوت Build متوالی، برنامه نویسی جفتی، توسعه آزمایش محور و غیره استفاده کنیم و البته چگونگی ترکیب این موارد با اسکرام را نیز بررسی کردیم.

آزمایش های ما در طی 1 سال یک فرآیند یادگیری پایدار بود که داستان به همین جا ختم نمی شد. من به این نتیجه رسیده بودم، برای اینکه این شرکت بتواند بهترین حالت اسکرام را نسبت به وضعیت منحصر به فرد خود پیاده سازی نماید باید در طول انجام اسکرام (طی جلسات بازبینی اسکرام) به این یادگیری پایدار ادامه دهد.

رفع مسئولیت

این کتاب ادعا نمی نماید که راه درست را برای انجام اسکرام نشان می دهد. و این کتاب فقط روش هایی را برای انجام اسکرام نشان می دهد که آنها نتیجه یک سال آزمایش و تحقیق پایدار ما بر روی استفاده از اسکرام بوده اند. حتی ممکن است شما به این نتیجه برسید که تمام این ها اشتباه بوده است. تمام مطالب ارائه شده نظرات شخصی من هستند و این به معنی اظهار و اعلان رسمی از طرف شرکت کریسپ (شرکت من) و یا مشتری فعلی من نمی باشد. به همین خاطر از ذکر عنوان هر نوع محصول و یا شخص پرهیز کرده ام.

چرا این را نوشتیم؟

زمانیکه من اسکرام را یاد می گرفتم، کتاب های اسکرام و Agile متعددی را خواندم، سایت ها و تالارهای گفتگوی اسکرام را دنبال می کردم، مدارک کن شوئبر را اخذ کردم، سوال هایم را از او می پرسیدم، وقت زیادی را با هم دانشگاهی ها بر روی اسکرام صرف می کردیم. یکی از بهترین

منابع یادگیری اطلاعات , داستان های جنگ واقعی دیگران می بود. داستان جنگ آنان تبدیل به اصول و روش ها می شد. به عبارت ساده تر , چگونگی روبرویی آنان با مشکل و روش مقابله با آن یک داستان جنگ خواننده می شود. آنها من را که یک تازه کار بودم با داستان ها جنگ خود تعلیم می دادند.

و هم اکنون فرصتی پیش آمده است که من درس هایی را پس بدهم . این داستان جنگ من است که می خوانید. امیدوارم که این کتاب بازخورد های سودمندی از جنگ های ما در اختیار شما قرار دهد.

اما اسکرام چیست ؟

آه , ببخشید . شما کاملاً با مفهوم اسکرام و XP نا آشنا هستید ؟ در این حالت بهتر است کمی با مبانی و تعاریف اسکرام آشنا شوید. می توانید از لینک های زیر برای این منظور استفاده کنید :

- <http://agilemanifesto.org>
- <http://www.mountangoatsoftware.com/scrum>
- <http://xprogramming.com/xpmag/whatisxp>

اگر علاقمند ادامه دادن به خواندن هستید نگران نباشید و احساس راحتی کنید و به خواندن ادامه دهید. بسیاری از مفاهیم اسکرام در ادامه به مرور توضیح داده شده اند .

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

2

چگونه بک لاگ محصول را ایجاد می کنیم

بک لاگ محصول به منزله قلب اسکرام می باشد و اینجا همان جایی هست که همه چیز شروع می شود. بک لاگ محصول به طور کلی یک لیست اولویت بندی شده از نیازمندی ها ، داستان ها ، ویژگی ها و یا هر مورد دیگری می باشد. مواردی که مشتری درخواست کرده است. و البته برای شرح این موارد از اصلاحات مشتری و اصطلاحات تجاری استفاده می شود .

ما این موارد را داستان ها³ و یا آیتم های بک لاگ می نامیم .

داستان های بک لاگ محصول ما شامل فیلد های زیر می باشند :

- **شماره :** یک شناسه یکتا برای هر داستان.
- **نام :** نام داستان که به صورت مختصر کار داستان را شرح می دهد. برای مثال "مشاهده تراکنش های شما". این نام به حد کافی واضح می باشد زیرا زمانی که توسعه گران و یا مالک محصول به این داستان نگاه می کنند می فهمند که دقیقاً درباره چه چیزی باید صحبت کنند و این داستان به اندازه کافی روشن است زیرا می توان این را از داستان های دیگر تمیز داد. معمولاً نام مناسب 2-10 کلمه می باشد.
- **اهمیت :** درجه اهمیت داستان از نظر مالک محصول. برای مثال 10 یا 150 و یا بالا = پر اهمیت و پایین = کم اهمیت.
- من از اصلاح "اولویت" در این مورد پرهیز کردم از اینرو که در این روش عدد 1 برای پر اهمیت ترین داستان ها انتخاب می شود اما زمانیکه شما می خواهید اهمیت این داستان را از قبل بیشتر نمایید چه عددی را انتخاب خواهید کرد ؟ 0 یا -1 ؟
- **برآورد اولیه :** نشان گر ارزیابی اولیه تیم از داستان می باشد و نمایانگر این است که این داستان نسبت به بقیه چقدر کار می طلبد. واحد این فیلد Story Point می باشد که در بعضی از موارد معادل "نیرو-روز" ایده آل⁴ خواهد بود.

³ Stories

- از تیم بپرسید: "اگر تعداد افراد مطلوب برای انجام این داستان موجود باشند و شما خود را همراه غذای مورد نیاز در یک اتاق حبس کنید تا این داستان را انجام بدهید، چند روز طول خواهد کشید تا داستان را تمام کنید؟" اگر آنها به چنین صورتی جواب دادند: "اگر ما سه نفر باشیم به طور حدودی در طی 4 روز می توانیم کار را تمام کنیم". پس برآورد اولیه 12 Story Point خواهد شد.
- لازم نیست که همه موارد را به صورت مطلق در نظر بگیرید (این داستان 2 Story Point است پس 2 روز طول خواهد کشید) مسئله مهم این است که شما باید در برآورد از نسبت داستان ها استفاده کنید (این یک داستان 2 Story Point ای است پس این نصف زمان یک داستان 4 Story Point ای طول خواهد کشید).
- **چگونگی دمو:** توضیح مختصر در مورد اینکه زمان ارائه دموی اسپرینت این داستان چگونه باید باشد. توضیح لزوماً باید ساده و قابل فهم باشد: "این را انجام بده، سپس آنرا، آنگاه باید آن انجام شده باشد".
- اگر از روش توسعه آزمایش محور (TDD) استفاده می نمایید این فیلد می تواند برای شما خیلی با ارزش باشد.
- **نکات:** هر توضیح دیگر، اطلاعات بیشتر یا رجوع به منابع دیگر جهت وضوح بیشتر داستان را می توانید به صورت خلاصه در این فیلد یادداشت کنید.

نمونه ای از بک لاگ محصول

ش	نام	اهمیت	برآورد	چگونگی دمو	نکات
1	سپرده گذاری	30	5	لاگین - باز کردن صفحه سپرده گذاری - سپرده گذاری مبلغ 10 دلار - بازکردن صفحه موجودی که باید موجودی 10 دلار اضافه شده باشد.	به دیاگرام توالی UML نیاز است. نیازی به استفاده از encryption نیست.
2	مشاهده تراکنش های شما	10	8	لاگین - کلیک بر روی تراکنش ها و مشاهده تراکنش آخر - بازکردن صفحه سپرده گذاری و انجام سپرده گذاری - باید این سپرده گذاری در لیست تراکنش ها نشان داده شود.	برای اینکه لیست زیاد کشیده نشود باید از خاصیت Paging استفاده شود. این صفحه باید مانند صفحه مشاهده کاربران طراحی شود.

فیلدهای زیادی را برای بک لاگ محصول مورد آزمایش قرار دادیم ، اما طی اسپرینت های مختلف این 6 فیلد پر کاربردترین فیلدهای ما بودند.

ما فایل بک لاگ محصول را در یک فایل Excel قابل ویرایش روی شبکه با فیلد های بالا ایجاد می کردیم. مالکیت و مدیریت این بک لاگ با مالک محصول می باشد ولی با این حال دسترسی به این فایل برای توسعه گران نیز امکان پذیر می باشد. شاید آنها بخواهند برای وضوح بیشتر یک داستان / توضیحاتی به آن اضافه نمایند و یا شاید برآورد آن را عوض نمایند .

به یک دلیل ما به جای قرار دادن این فایل بر روی مخزن سورس کنترل آن را بر روی یک درایو اشتراک شده قرار می دهیم. به این دلیل که در سورس کنترل با تغییر فایل مربوطه احتمال بروز مشکلات یا کانفلیکت و یا قفل فایل برای دیگران به وجود خواهد آمد.

البته تمام ابزارها و موارد دیگر بر روی مخزن سورس کنترل قرار می گیرند.

فیلد های اضافی برای داستان ها

گاهها از فیلد های اضافی در داخل بک لاگ محصول استفاده می کنیم که اکثرا به خاطر ایجاد سهولت در اولویت بندی داستان ها برای مالک محصول می باشند.

- **دسته :** برای دسته بندی داستان ها استفاده می شود . مثلا دسته "بهینه سازی" مالک محصول می تواند تمام داستان های مربوط به بهینه سازی را فیلتر کند و مثلا اولویت پیاده سازی آنها را پایین در نظر بگیرد.

- **اجزاها :** این فیلد معمولا در فایل Excel به صورت یک دسته چک باکس در نظر گرفته می شود. برای مثال اجزای "دیتابیس - سرور - کلاینت". در این حالت تیم ها و یا مالک محصول می توانند در انتخاب داستان ها با یکدیگر دقت بیشتری داشته باشند. معمولا این فیلد زمانی مفید است که شما چند تیم اسکرام دارید برای مثال تیم سرور و تیم کلاینت. تیم کلاینت در این حالت داستان های مربوط به خود را انتخاب می کند و تیم سرور هم همینطور.

- **درخواست کننده :** اگر مالک محصول بخواهد بداند که کدام مشتری و یا کدام ذینفع این داستان را درخواست داده است تا بتواند روند پیشرفت داستان را به او اعلام نماید می تواند از این فیلد نیز استفاده کند.

- **شماره پیگیری باگ** : اگر از یک سیستم جداگانه برای پیگیری باگ ها استفاده می کنید مانند ما که از Jira استفاده می کنیم ، این بسیار سودمند می تواند باشد که ارتباط مستقیمی از طریق نگه داری شماره باگ در فایل Excel به وجود آورید.

چگونه بک لاگ محصول را در سطح تجاری نگه می داریم

اگر مالک محصول پیش زمینه فنی داشته باشد ممکن است چنین داستان هایی را به لیست داستان ها اضافه کند "اضافه کردن ایندکس ها به جدول رویداد ها". چرا او چنین درخواستی دارد؟ حقیقت امر این است که شاید هدف او از خواستن چنین چیزی فقط "افزایش سرعت جستجو در فرم رویداد ها" باشد.

شاید اصلا کندی جستجو ربطی به بحث نگه داری ایندکس ها در جدول رویدادها نداشته باشد و شاید این مورد علت دیگری دارد. بهتر است که وظیفه حل مشکلات به تیم واگذار شود و تیم دنبال این باشد که چگونه باید مشکلی حل شود و مالک محصول باید بر روی اهداف تجاری متمرکز باشد.

زمانی که من با چنین داستان های فنی مواجه می شوم ، به طور معمول از مالک محصول می پرسم "به چه دلیل". از این طریق به هدف پشت پرده مالک محصول پی می بریم و به جای داستان "اضافه کردن ایندکس ها به جدول رویداد ها" از داستان "افزایش سرعت جستجو در فرم رویداد ها" استفاده می کنیم.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

3

چگونه برای برنامه ریزی اسپرینت آماده می شویم

بسیار خوب، روز برنامه ریزی اسپرینت بسیار سریع فرا رسید.

قبل از جلسه برنامه ریزی اسپرینت مطمئن شوید که بک لاگ محصول تمیز و مرتب باشد. معنی این چیست؟

- بک لاگ محصول باید موجود باشد.
- باید به ازای هر بک لاگ محصول یک مالک محصول وجود داشته باشد (برای هر محصول).
- تمام آیتم های مهم باید رتبه اهمیت داشته باشند (درجه اهمیت)، رتبه های اهمیت متفاوت.
 - در واقع، برای آیتم های کم اهمیت می توان یک رتبه اهمیت در نظر گرفت زیرا احتمالاً آنها در جلسات برنامه ریزی اسپرینت مطرح نخواهند شد.
 - هر داستانی که مالک محصول اعتقاد دارد باید در اسپرینت بعدی انجام شود باید درجه اهمیت یکتایی داشته باشد.
 - درجه اهمیت داستان ها فقط برای مرتب کردن آنها بر اساس اهمیت آنها می باشند. پس اگر داستان A درجه اهمیت 20 را داشته باشد و داستان B درجه اهمیت 100، به این معنی خواهد بود که B بسیار مهم تر از A می باشد و به این معنی نیست که B 5 برابر مهم تر از A می باشد. اگر درجه اهمیت B 21 بود باز به همین معنی خواهد بود که B مهم تر از A می باشد.
 - بهتر است بین درجه اهمیت ها کمی فاصله بگذارید تا بتوانید داستان های دیگر را مابین آنها قرار بدهید. در مثال بالا بدلیل وجود نداشتن فاصله فقط می توان برای C درجه اهمیت 20.5 را در نظر گرفت که زیاد مناسب نمی باشد.
- مالک محصول باید بتواند همه داستان های بک لاگ محصول را درک نماید. (درست است که داستان ها را خود مالک می نویسد ولی شاید در بعضی از اوقات برنامه نویسان داستان

هایی را به لیست اضافه کنند که این ها را نیز باید مالک اولویت بندی نماید). نیاز نیست که مالک محصول دقیقا بداند که چه چیزی نیاز به پیاده سازی دارد ولی او باید بداند که چرا داستان در لیست بک لاگ قرار دارد.

نکته: افرادی به جز مالک محصول می توانند داستان هایی را به لیست بک لاگ محصول اضافه کنند ولی تنها کسی که می تواند درجه اهمیت آن را مشخص کند شخص مالک محصول می باشد و در ضمن تنها کسی که می تواند برآورد اولیه هر داستان را تعیین کند فقط تیم (ها) می باشد.

روش های دیگری که ما امتحان و یا ارزیابی کرده ایم:

- استفاده از Jira (سیستم پیگیری باگ ما) به عنوان نگه دارنده بک لاگ محصول. بسیاری از مالکان محصول ما این برنامه را بسیار گسترده دانستند. برنامه Excel برای ایجاد و دستکاری بک لاگ خوب و راحت می باشد. به راحتی می توانید به آن ستون اضافه کنید، خانه های آن را رنگی کنید، آیتم ها را مرتب کنید، خروجی و ورودی بگیرید و غیره.
- استفاده از ابزار های پشتیبان فرآیند Agile مانند VersionOne, ScrumWorks, XPlanner و غیره. ما هیچ یک از ابزار ها موجود را نتوانستیم آزمایش کنیم ولی احتمالا این کار را انجام خواهیم داد.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

4

چگونه برنامه ریزی اسپرینت را انجام می دهیم

برنامه ریزی اسپرینت یک جلسه حساس و مهم می باشد و احتمالاً مهم ترین رویداد در اسکرام به شمار می رود (البته در نظر شخصی من). یک جلسه برنامه ریزی اسپرینت بد می تواند باعث از بین رفتن کل اسپرینت شود.

هدف از جلسه برنامه ریزی اسپرینت این است که به تیم اطلاعات کافی در مورد بخشی از محصول داده شود تا آنها بتوانند در آرامش چند هفته ای بر روی بخشی از محصول کار نمایند.

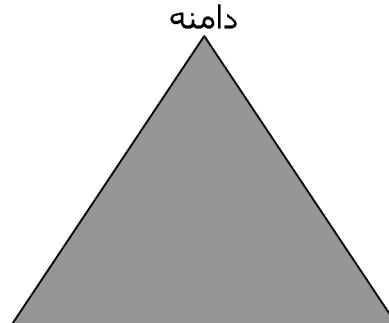
بسیار خوب ، شاید تعریف بسیار مبهم به نظر می آید. خروجی واقعی جلسه برنامه ریزی اسپرینت عبارتست از :

- هدف اسپرینت
- لیست اعضای تیم (و سطح مسئولیت آنها اگر آنها به صورت 100٪ حضور ندارند)
- بک لاگ اسپرینت (= لیست داستان هایی که در اسپرینت هستند)
- روز مشخص شده برای ارائه دمو اسپرینت
- زمان و مکان تعیین شده برای انجام جلسات روزانه اسکرام

چرا مالک محصول باید حضور داشته باشد

بعضی از اوقات مالک محصول برای حضور نداشتن در جلسه برنامه ریزی اسپرینت با اکراه می گوید "دوستان ، من هر چیزی را که می خواستم لیست کرده ام و وقت این را ندارم در جلسه برنامه ریزی شما باشم". و این واقعا یک مشکل جدی می باشد.

دلیل اینکه چرا باید کل تیم و مالک محصول در جلسه برنامه ریزی اسپرینت حضور داشته باشند این است که: هر داستان شامل سه متغیر می شود که تمام متغیرها به یکدیگر وابسته هستند.



برآورد

اهمیت

متغیرهای دامنه و اهمیت توسط مالک محصول تعیین می شوند. متغیر برآورد نیز توسط تیم تعیین می شود. در یک جلسه برنامه ریزی اسپرینت این سه متغیر به طور متوالی طی صحبت های چهره-به-چهره بین تیم و مالک محصول نسبت به یکدیگر میزان سازی می شوند.

به طور معمول مالک محصول با معرفی هدف اسپرینت به صورت خلاصه و معرفی داستان های مهم، جلسه را شروع می کند. بعد تیم شروع به برآورد زمانی هر داستان می کند. نخست داستان های مهم انتخاب می شوند. برای فهم بیشتر داستان تیم از متغیر دامنه شروع می کند. "در داستان "حذف کاربر" آیا تراکنش های در حال اجرا و یا معلق کاربر را باید حذف نماید؟" گاهی جواب مالک محصول، تیم ها را غافل گیر می کند زیرا که باید برآورد اولیه خود را تغییر دهند.

در بعضی از موارد، برآورد زمانی یک داستان آن چیزی نمی شود که مالک محصول انتظار داشت پس او شاید درجه اهمیت داستان را تغییر بدهد. یا شاید دامنه داستان را تغییر بدهد که در این حالت تیم باید دوباره داستان را برآورد نماید.

اینگونه همکاری های مستقیم بنیاد اسکرام را تشکیل می دهند و در واقع همان بنیاد توسعه نرم افزار چابک (Agile Software Development).

اگر مالک محصول باز مایل به شرکت در جلسه برنامه ریزی اسپرینت نباشد، آن موقع چه کاری باید انجام دهیم؟ من معمولاً در این مواقع از یکی از روش های زیر استفاده می کنم:

- سعی کنید لزوم اهمیت شرکت او را در جلسات برنامه ریزی به وی بفهمانید و امیدوار باشید که ذهنیت او را تغییر داده باشید.
- سعی کنید یکی از اعضای تیم را به صورت داوطلبانه به عنوان نماینده مالک محصول انتخاب نمایید و به مالک محصول بگویید تا زمانیکه شما نمی توانید حضور داشته باشید، "جف" نماینده شما برای تیم می باشد. و او این حق را دارد که اولویت بندی و دامنه داستان ها را تغییر بدهد. جف باید قبل از جلسه، اطلاعات خود را با مالک محصول همسان سازی نماید. اگر جف که داوطلب نمایندگی مالک محصول بود را مناسب نمایندگی نمی دانید بهتر است کس دیگری را در نظر بگیرید.
- سعی کنید تیم مدیریتی را قانع نمایید تا مالک محصول جدیدی را برگزینند.
- جلسه برنامه ریزی را تا زمانیکه مالک محصول وقت داشته باشد به تعویق بیندازید. و ضمناً در آن مدت از قبول انجام هر کاری سر باز زنید و به اعضای تیم اجازه بدهید تا هر کاری که صلاح می دانند در آن زمان انجام بدهند.

چرا کیفیت بحث کردنی نیست؟

در مثلث بالا من عمداً از ذکر متغیر چهارم یعنی "کیفیت" خودداری کردم.

وجه تمایز بین کیفیت داخلی و کیفیت خارجی:

- کیفیت خارجی شامل مواردی می شود که برای کاربر سیستم قابل درک هست. یک رابط کاربری کند و یا غیر بصری، نمونه ای از کیفیت خارجی ضعیف می باشد.
- کیفیت داخلی به مواردی اطلاق می شود که معمولاً برای کاربر سیستم قابل مشاهده نیستند اما عمیقاً بر روی قابلیت نگه داری سیستم تاثیر می گذارند. مواردی مانند طراحی سیستم سازگار، پوشش تست بر روی کدها، خوانایی کدها، رفاکتور کردن و غیره.

یک سیستم با کیفیت داخلی بالا می تواند کیفیت خارجی ضعیفی داشته باشد. اما یک سیستم با کیفیت داخلی ضعیف به ندرت می تواند دارای کیفیت خارجی بالایی باشد. بسیار سخت است که بر روی یک پایه پوسیده یک اثر زیبا بنا کنید.

من معمولا کیفیت خارجی را به عنوان بخشی از دامنه داستان در نظر می گیرم. در بعضی از موارد ممکن است در ارائه دمو یک اسپرینت رابط کاربری آن خوب کار نشده باشد ولی در ارائه دوم این مورد کار شده باشد و کیفیت خارجی آن ارتقاء یافته باشد. من این مورد را به خود مالک محصول واگذار می کنم تا او تصمیم بگیرد که ارتقاء سطح کیفیت خارجی در چه اسپرینتی انجام شود. با ارتقاء سطح کیفیت خارجی دامنه نیز افزایش خواهد یافت.

اما کیفیت داخلی قابل بحث نیست که در این اسپرینت انجام شود یا در بعدی. این وظیفه تیم است که کیفیت داخلی سیستم را همیشه در حد قابل نگه داری حفظ نمایند و به طور خلاصه کیفیت داخلی قابل بحث نیست. (تقریبا هیچ وقت)

چگونه در مورد فرق کیفیت داخلی و کیفیت خارجی با مالک محصول سخن می گوئیم؟

فرض کنید مالک محصول چنین می گوید "بسیار خوب دوستان، من به برآورد 6 Story Point شما احترام می گذارم، اما اگر شما تماما تلاش کنید می توانید چنین کاری را سریعا در نصف زمان برآورد شده حل نمایید".

آها! او تلاش می کند که کیفیت داخلی را به عنوان یک متغیر استفاده نماید. چگونه این را فهمیدم؟ زیرا او از ما می خواهد که برآورد داستان را تغییر بدهیم بدون اینکه هزینه اش را پردازد؟ هزینه این تغییر فقط کاستن از دامنه می باشد. "سریعا حل نمایید" را باید به عنوان یک زنگ خطر در ذهن خود ثبت نمایید.

و چرا ما به چنین درخواستی جواب نمی دهیم؟

تجربه من این است که قربانی کردن کیفیت داخلی تقریبا همیشه ایده وحشتناکی بوده است. کدی که یکباره بدون دقت و به خرابی شروع شود برگرداندن کیفیت به آن کد بسیار سخت خواهد شد.

به جای این بحث را به سمت دامنه هدایت می کنم. "از آنجایی که این داستان برای شما اهمیت زیادی دارد آیا می توانیم دامنه داستان را کاهش دهیم تا پیاده سازی زودتر انجام شود؟" شاید ما می توانیم

هندلینگ خطاها را به صورت ساده در این داستان انجام دهیم و هندلینگ پیشرفته را به داستان دیگر وانهم؟ یا آیا می توانیم اولویت داستان های دیگر را کاهش دهیم تا به این یکی داستان متمرکز شویم؟

مالک محصول در این حالت یاد خواهد گرفت که کیفیت داخلی قابل بحث نیست و باید این قابلیت را پیدا کند که در این حالت از متغیر های دیگر استفاده نماید.

جلسه برنامه ریزی اسپرینت که کشیده می شود ...

بیش ترین مشکل درباره جلسه برنامه ریزی اسپرینت این است که :

- افراد گمان نمی کنند که این جلسات زیاد طول خواهند کشید
- ... ولی آنها زیاد طول می کشند

همه چیز در اسکرام زمانبندی شده است. من این ویژگی را دوست دارم. قانون ساده و پایدار. ما سعی می کنیم که در اسکرام به این قانون بچسبیم.

پس زمانی که در جلسه برنامه ریزی اسپرینت به اواخر جلسه نزدیک می شویم و هنوز خبری از هدف اسپرینت و یا بک لاگ اسپرینت نیست ، چه باید بکنیم؟ آیا باید جلسه در همان لحظه قطع شود؟ آیا می توانیم یک ساعت دیگر به جلسه اضافه کنیم؟ آیا ما می توانیم پس از اتمام زمان ، جلسه را به فردا موکول کنیم؟

این مورد پی در پی بخصوص برای تیم های تازه کار اتفاق می افتد. شما در این مورد چه کاری انجام می دهید؟ ما چه کاری انجام می دهیم؟ من با بیرحمی جلسه را کوتاه می کنم و پایان جلسه را اعلام می کنم. به تیم و مالک محصول می گویم که " این جلسه بعدا از 10 دقیقه تمام خواهد شد. ما در واقع زمان زیادی برای برنامه ریزی اسپرینت نداریم و باید اسپرینت را با چیزهایی که داریم انجام بدهیم یا باید برای برنامه ریزی اسپرینت ، 4 ساعت دیگری زمانبندی کنیم که فردا صبح ساعت 8 چطور است؟". شما می توانید حدس بزنید که جواب آنها چه خواهد بود (O : .

آن موقعی که چیزی تکمیل نمی شود معمولا افراد خسته هستند و من سعی می کنم اجازه بدهم تا جلسه به فردا کشیده شود. زمانیکه آنها طی جلسه 2-8 ساعته نتوانسته اند طرح اسپرینت را برنامه ریزی کنند اضافه کردن یک ساعت احتمالا برای مدیریت کارها موثر نخواهد بود.

من در این حالت جلسه را کوتاه و قطع می نمایم. برعکس جلسه روز قبل ، در جلسه روز بعد تیم با توجه به درسهای یادگرفته ، موثر تر عمل خواهد کرد. بعلاوه افراد زمانیکه شما مدت جلسه را اعلام می کنید کمتر مقاومت خواهند کرد .

یاد بگیرید که زمانبندی های خود را حفظ نمایید ، یاد بگیرید که مدت زمان واقع گرایانه برای زمانبندی های خود تعیین کنید. این مورد هم برای طول جلسه برنامه ریزی و هم برای طول اسپرینت تاکید می شود.

دستور جلسه برنامه ریزی اسپرینت

داشتن یک دستور جلسه و برنامه اولیه برای جلسه برنامه ریزی اسپرینت ریسک شکسته شدن زمانبندی را کاهش خواهد داد.

زمانبندی نمونه برای این جلسه :

جلسه برنامه ریزی اسپرینت : 13:00 تا 17:00 (10 دقیقه استراحت در هر ساعت)

- **13:00 – 13:30** . مالک محصول هدف اسپرینت را به طور خلاصه شرح می دهد و داستان های مربوط به آن را مشخص می کند. زمان و مکان دمو تعیین می شود.
- **13:30 – 15:00** . تیم در صورت نیاز آیتم ها را به آیتم های کوچک تر می شکند. مالک محصول در صورت نیاز درجه اهمیت آنها را تغییر می دهد. آیتم ها واضح تر می شوند. فیلد "چگونگی دمو" برای تمام موارد با اهمیت پر می گردد.
- **15:00 – 16:00** . تیم داستان های اسپرینت را انتخاب می کند. سرعت تیم بر اساس واقعیت سنجیده می شود.

• **16:00 – 17:00** . زمان و مکان جلسه روزانه اسکرام انتخاب می شود (اگر از آخرین اسپرینت متفاوت باشد). در نهایت داستان ها به وظایف شکست داده می شوند.

دستور جلسه نمونه بدین معنی نیست که دقیقا به زور جلسه را اینگونه برگزار کنید. مدیر اسکرام شاید جلسه را مدت دارتر و یا کوتاه تر طی زمانبندی های کوچک تر در صورت نیاز برای پیشرفت جلسه برگزار کند.

تعیین طول اسپرینت

یکی از خروجی های جلسه برنامه ریزی اسپرینت روز مشخص شده برای ارائه دمو اسپرینت می باشد. این بدین معنی می باشد که شما مجبورید درباره طول اسپرینت تصمیم بگیرید .

پس طول یک اسپرینت خوب چقدر است؟

خوب ، اسپرینت های کوتاه خوب هستند. آنها اجازه می دهند تا شرکت Agile باشد. اسپرینت کوتاه = چرخه بازخورد کوتاه = ارائه های مکرر = بازخورد های مکرر مشتری = صرف شدن زمان کم بر روی کج روی ها = یادگیری و بهبود سریع تر و غیره .

اما ، اسپرینت های بلند نیز خوب هستند. تیم زمان بیشتری برای ساخت قسمت های مختلف دارد. فضای بیشتری برای برگشتن و احیاء اشتباهات خود در حین ساخت خواهند داشت. در کل خلف وعده کمتری نسبت به وعده های داده شده در جلسه برنامه ریزی اسپرینت و ارائه دمو انجام خواهد شد.

به طور کلی مالک محصول اسپرینت های کوتاه و توسعه گران اسپرینت های بلند را دوست دارند. پس طول اسپرینت قابل توافق خواهد بود. طبق تجربه ما مورد علاقه ترین طول برای اسپرینت 3 هفته می باشد. بیشتر تیم های ما (نه همه) از اسپرینت های 3 هفته ای استفاده می کنند. کوتاه به اندازه کافی که بتوانیم به Agile دست بیاییم و بلند به اندازه کافی برای تیم که بتوانند از اشتباهات به وجود آمده در طی اسپرینت احیا شوند.

ما به یک نتیجه در مورد طول اسپرینت رسیدیم و اینکه طول اسپرینت را باید بر اساس تجربه اسپرینت های قبلی انتخاب کنید. در اسپرینت های نخست وقت خود را بر روی آنالیز طول اسپرینت هدر ندهید. یک زمان معقول برای یک یا دو اسپرینت نخست انتخاب کنید. سپس در اسپرینت های بعدی در صورت نیاز طول اسپرینت را تغییر دهید.

به هر صورت شما باید به یک طول ایده آل اسپرینت طی تجربه برسید و به آن طول اسپرینت بچسبید. بعد از چند ماه به این تجربه دست یافتیم که طول 3 هفته برای ما عالی می باشد. پس ما از این طول 3 هفته استفاده کردیم. بعضی ها اعتقاد داشتند که این خیلی بلند است و بعضی اعتقاد داشتند این طول خیلی کوتاه است. ولی ما با حفظ این طول مانند یک ضربان یکسان عمل می کردیم که همه در این ضربان راحت بودند. هیچ بحثی بر روی زمان ارائه دمو نبود. زیرا همه می دانستند که بعد از هر 3 هفته زمان ارائه دمو خواهد بود.

تعیین هدف اسپرینت

در یک زمان مشخص طی جلسه برنامه ریزی اسپرینت سوال می پرسیم که "هدف از این اسپرینت چه می باشد؟" و همه فقط به من خیره می شوند و مالک محصول هم شروع به خاریدن چانه خود می کند. این معمولا برای تیم های اسکرام اتفاق می افتد.

به دلایلی، مشخص کردن هدف اسپرینت سخت می باشد. اما من به این باور رسیده ام که داشتن هدف نصفه و نیمه بهتر از نداشتن هدف می باشد. یک هدف می تواند مواردی مانند این ها باشد "در آوردن پول بیشتر" یا "کامل کردن 3 داستان اولویت بالا" یا "تحت تاثیر قرار دادن مدیر عامل" یا قرار دادن سیستم در وضعیت خوب جهت ارائه نسخه بتا به صورت آنلاین" یا هر چیز دیگر. مهمترین مسئله این است که در تعیین هدف باید از اختصارات و کلمات تجاری به جای علائم فنی استفاده شود. بدین معنی که افراد خارج از تیم بتوانند آن را درک نمایند.

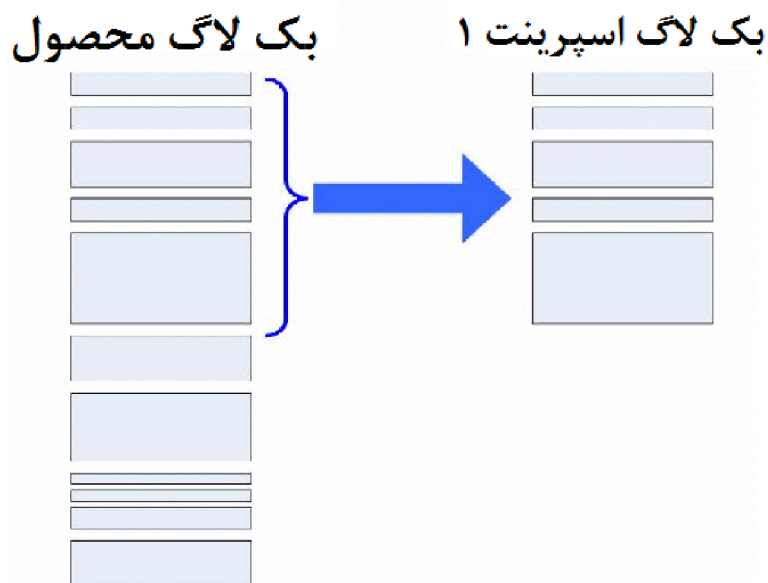
هدف اسپرینت باید بتواند به سوال بنیادین "چرا ما باید این اسپرینت را انجام بدهیم؟ چرا به جای این، وقت خود را در تعطیلات صرف نکنیم؟" جواب بدهد. در واقع یک راه برای دریافت هدف اسپرینت از مالک محصول، پرسش سوال هایی از این قبیل می باشد.

هدف اسپرینت باید چیزی باشد که قبلا به آن دست نیافته ایم. "تحت تاثیر گذاشتن مدیر عامل" هدف خوبی است اما اگر قبلا به چنین هدفی دست یافته ایم، پس اینبار موثر و با ارزش نخواهد بود. در این حالت که مدیر عامل قبلا تحت تاثیر قرار گرفته است همه می توانند به خانه هایشان بروند و هدف را دست یافته ببینند.

هدف اسپرینت معمولا در طی برنامه ریزی اسپرینت احمقانه به نظر می رسد. ولی این هدف معمولا در نیمه اسپرینت به درد خواهد خورد. زمانیکه افراد دچار سردرگمی خواهند شد که چه کاری را باید انجام دهند. آنها خواهند توانست از هدف اسپرینت مانند یک راهنمای کامل استفاده کنند. اگر چندین تیم اسکرام دارید که آنها بر روی محصولات مختلف کار می کنند می توانید هدف های تیم های مختلف را بروی یک ویکی مشترک قرار دهید تا در شرکت هر کسی در هر سطحی بداند که هم اکنون اهداف شرکت به چه صورتی است.

چه داستان هایی باید در اسپرینت باشند

یکی از فعالیت های اصلی در جلسه برنامه ریزی اسپرینت، تصمیم گیری در مورد قرارگیری داستان هایی در بک لاگ اسپرینت می باشد. به عبارتی کدام یک از داستان های داخل بک لاگ محصول باید به داخل بک لاگ اسپرینت کپی شوند؟



به تصویر بالا نگاه کنید. هر یک از مستطیل‌ها نمایانگر یک داستان می‌باشد که بر اساس اهمیت اشان مرتب شده‌اند. مهم‌ترین داستان در بالای لیست قرار گرفته است. اندازه هر مستطیل نمایانگر اندازه آن داستان می‌باشد (مثلاً برآورد زمانی داستان بر اساس Story Point). اندازه آکولاد نمایانگر سرعت برآورد شده تیم می‌باشد. به طور ساده یعنی اینکه تیم چند Story Point می‌تواند طی اسپرینت بعدی تکمیل نماید.

بک لاگ اسپرینت که در طرف راست عکس قبلی قابل مشاهده است شامل تصویری از بخشی از بک لاگ محصول می‌باشد. بک لاگ اسپرینت لیستی از داستان‌هایی می‌باشد که تیم متعهد شده است که آنها را در این اسپرینت انجام دهد.

تیم تصمیم می‌گیرد که چه مقدار داستان را در اسپرینت انجام بدهد و مالک محصول و یا هیچ‌کس دیگر حق دخالت در این بخش را ندارند.

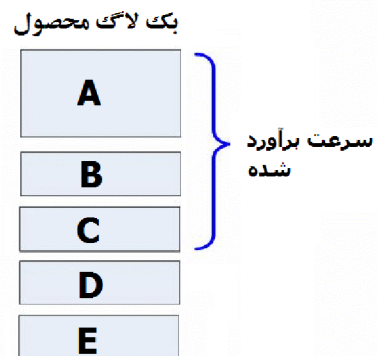
این مسئله باعث به وجود آمدن 2 سوال اساسی می‌شود:

1. تیم چگونه تصمیم می‌گیرد که کدام داستان‌ها در اسپرینت قرار بگیرند؟
2. مالک محصول چگونه می‌تواند بر روی تصمیم آنها تاثیر بگذارد؟

با سوال دوم شروع می‌کنم.

مالک محصول چگونه می‌تواند در انتخاب داستان‌های اسپرینت تاثیر گذار باشد؟

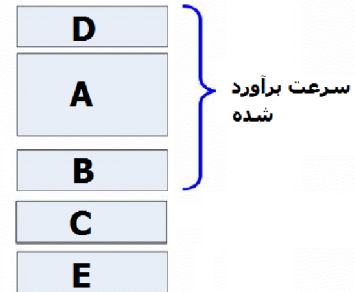
وضعیت زیر را در نظر بگیرید که ما در جلسه برنامه ریزی اسپرینت چنین وضعیتی داریم:



مالک محصول از اینکه داستان D قرار نیست در این اسپرینت انجام شود ناکام مانده است. انتخاب او برای کام‌روایی در جلسه برنامه ریزی اسپرینت چه چیز می‌تواند باشد؟

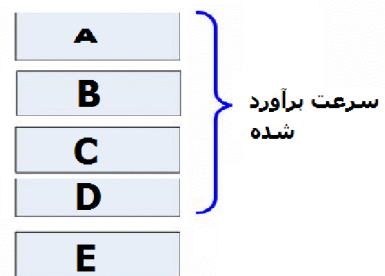
یک انتخاب او می تواند این باشد که اولویت بندی ها را تغییر دهد (دوباره الویت بندی کند). اگر او اولویت داستان D را بالاتر از همه قرار بدهد آن داستان می تواند داخل اسپرینت قرار گیرد.

گزینه ۱



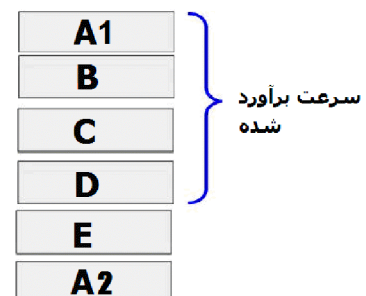
راه حل دوم تغییر دامنه می باشد. مالک محصول می تواند با تغییر دامنه داستان A باعث داخل شدن داستان D به لیست بک لاگ اسپرینت شود.

گزینه ۲



راه حل سوم تکه سازی داستان ها می باشد. مالک محصول می تواند داستان A را به چند تکه کوچک تر تقسیم کند تا به داستان D اجازه بدهد تا در اسپرینت قرار بگیرد.

بک لاگ محصول



همانطور که مشاهده کردید درست است که مالک محصول در انتخاب داستان برای اسپرینت دخالت مستقیم ندارد ولی می تواند با روش های مختلفی در این فرآیند نقش مهمی داشته باشد.

تیم چگونه تصمیم می گیرد که کدام داستان ها در اسپرینت قرار بگیرند؟

ما برای این منظور از دو تکنیک زیر بهره می بریم:

1. تجربه و احساس
2. محاسبه سرعت یا Velocity

برآورد توسط تجربه و احساس

مدیر اسکرام: "دوستان، آیا ما می توانیم داستان A را در این اسپرینت تکمیل نماییم؟" (اشاره به مهمترین داستان بک لاگ محصول).

لیزا: "البته که می توانیم این را طی اسپرینت 3 هفته ای انجام بدهیم".

مدیر اسکرام: "بسیار خوب، در مورد داستان B چطور؟" (اشاره به دومین داستان پر اهمیت).

لیزا و تام در هم آوایی: "هیچ مانع ای وجود ندارد".

مدیر اسکرام: "پس در مورد داستان های A, B و C چطور؟".

سم (به مدیر اسکرام): "آیا داستان C شامل هندلینگ پیشرفته خطاها هم می باشد؟".

مالک محصول: "نه، شما می توانید این را فعلا نادیده بگیرید و این فقط شامل هندلینگ ساده خطاها می باشد".

سم: "خوب پس داستان C می تواند خوب باشد".

مدیر اسکرام: "خوب پس در مورد داستان D چطور؟".

لیزا: "اهمممم...".

تام: "به نظرم ما می توانیم این را نیز انجام بدهیم".

مدیر اسکرام: "با اطمینان چند درصدی؟ 90٪ یا 50٪؟".

لیزا و تام: "در حدود 90٪".

مدیر اسکرام: "D در لیست قرار گرفت اما در مورد E چطور؟".

سم: "شاید".

مدیر اسکرام: "90٪ یا 50٪".

سم: "حدود 50٪".

لیزا: "من شک دارم".

مدیر اسکرام: "پس این مورد را کنار می گذاریم. ما متعهد می شویم که داستان های A,B,C,D را انجام بدهیم. ولی اگر توانستیم حتما E را هم انجام خواهیم داد ولی هیچ کس بر روی انجام شدن E حساب نکند فعلا. نظر شما چیست؟"

همه: "بسیار خوب".

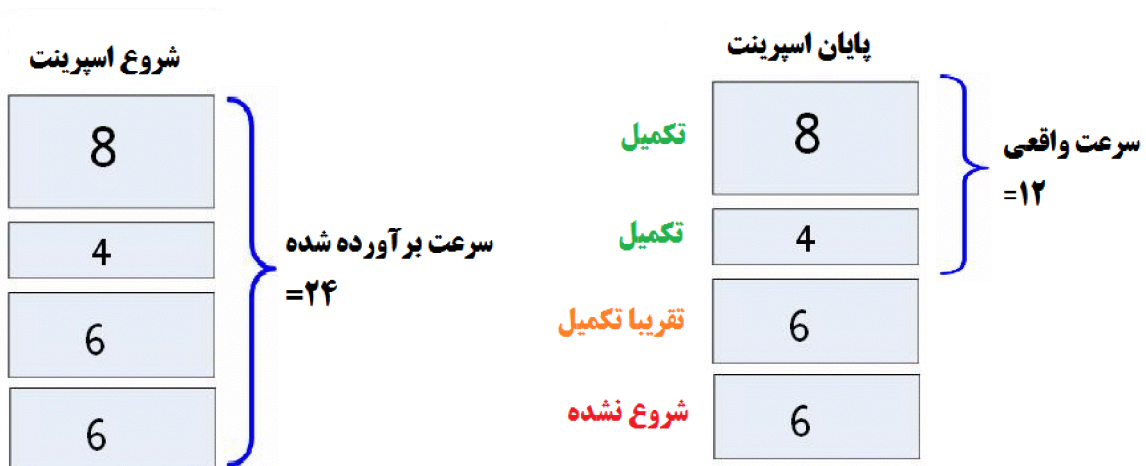
برآورد براساس تجربه و احساس در تیم های کوچک و برای اسپرینت های کوتاه موثرتر می باشد.

برآورد بوسیله محاسبه سرعت

این تکنیک شامل دو مرحله می شود:

1. تعیین سرعت برآورد شده.
2. محاسبه تعداد داستانی که شما می توانید بدون تجاوز از سرعت برآورد شده اضافه کنید.

سرعت یا Velocity عبارتست از اندازه گیری "میزان کار انجام شده" , این در حالی می باشد که هر آیتم دارای وزن مخصوص به خود است که در فیلد برآورد اولیه توسط تیم تعیین شده اند. عکس زیر یک مثال در مورد برآورد سرعت می باشد. هر یک از مستطیل ها یک داستان و عدد داخل آن نمایانگر برآورد اولیه داستان می باشد.



نکته: سرعت واقعی براساس جمع برآورد های اولیه داستان های تکمیل شده بدست آمده است. هر تغییری بر روی زمان برآورد داستان های تکمیل شده در طی اسپرینت نادیده گرفته خواهد شد.

شاید شما چنین بگویید: "کجای این اختلاف بدست آمده مفید است؟ بالا بودن و یا پایین بودن سرعت ممکن است به چندین عامل وابسته باشد. برنامه نویسان نفهم , برآورد اولیه نادرست , تجاوز از دامنه^۵ , اختلالات برنامه ریزی نشده در طی اسپرینت و غیره!".

من هم با شما موافقم , شاید این عدد اختلاف عدد خامی باشد , ولی باز می تواند مفید واقع شود. بدون توجه به نتیجه بدست آمده , این عدد به طور حدودی نمایانگر اختلاف آن مقدار کاری می باشد که ما فکر می کردیم می توانیم انجام بدهیم با آن مقدار کاری که انجام داده ایم.

اما در مورد داستانی که "تقریباً تکمیل" انجام شده است چطور؟ آیا می توانیم مثلاً نصف امتیاز آن را به عنوان کار شده بر روی سرعت واقعی لحاظ کنیم؟ در واقع این اصل اسکرام (و در واقع اصل Agile و تولید Lean) بسیار اضطراب آور است که تکمیل شدن آن داستان به این معنی می باشد که آن داستان قابل ارائه باشد. امتیاز داستان نصف-تکمیل شده برابر با صفر می باشد(در واقع منفی). می توانید برای مطالعه بیشتر کتاب "مدیریت کارخانه طراحی"^۶ "دونالد راینرتسن"^۷ و یا یکی از کتاب های پاپن دایک^۸ را مطالعه کنید.

پس توسط چه جادوی محرمانه ای ما سرعت را برآورد می کنیم؟ یکی از ساده ترین روش های برآورد سرعت رجوع به پیشینه تیم می باشد. سرعت آنها در چند اسپرینت قبلی چقدر بوده است؟ پس فرض کنید که سرعت آنها در اسپرینت بعدی هم در همان حدود اسپرینت های قبلی خواهد بود.

این تکنیک با عنوان "آب و هوای دیروز"^۹ شناخته می شود. این روش برای تیم هایی امکان پذیر می باشد که قبلاً چند اسپرینت را انجام داده باشند. (و آمارشان در دست باشد) و البته کار با روش و شرایط قبلی انجام شود. اندازه تیم همان اندازه قبلی باشد , شرایط کار فرق نکرده باشد , نفرات فرق نکرده باشند و غیره. و البته این روش برای همه مواقع مناسب نیست.

⁵ Scoop Creep

⁶ Managing the design factory

⁷ Donal Reinertsen

⁸ Poppendieck

⁹ Yesterday's Weather

نخست باید محاسبه آسانی در مورد منابع انسانی داشته باشیم. به این مثال توجه کنید: ما در حال برنامه ریزی یک اسپرینت 3 هفته ای هستیم (15 روز) همراه با 4 نفر در تیم. لیزا 2 روز به مرخصی خواهد رفت. دیو 50٪ حضور مستمر خواهد داشت و یک روز هم مرخصی. همه این ها را جمع می کنیم:

روزهای موجود	
15	لیزا
13	تام
15	سم
7	دیو
جمعا 50 نیرو - روز	

محاسبه این را به ما نشان داد که ما جمعا 50 نیرو - روز برای این اسپرینت موجود داریم.

آیا این برآورد سرعت ما بود؟ خیر. به این خاطر که واحد سرعت ما Story Point است و نه نیرو - روز. این 50 نیرو - روز را اصلاحا 50 نیرو- روز ایده آل می خوانند زیرا هیچ کار و یا اختلال برنامه ریزی نشده ای در محاسبه این عدد دخالت نداشته است.

و بدون شک در برآورد سرعت این عدد کمتر از 50 خواهد بود. اما چه قدر کمتر؟ ما برای این از اصلاح "ضریب توجه" استفاده می کنیم.

برآورد سرعت این اسپرینت:

(نیرو - روز موجود) × (ضریب توجه) = سرعت برآورد شده

ضریب توجه r برآورد میزان توجه تیم می باشد. یک ضریب توجه پایین نمایانگر وجود اختلالات زیاد در محیط تیم و یا برآوردهای خوش بینانه می باشد.

بهترین روش برای محاسبه ضریب توجه معقول توجه به اسپرینت قبلی می باشد. (یا معدل اسپرینت های قبلی).

محاسبه ضریب توجه اسپرینت قبلی :

$$\text{ضریب توجه} = (\text{سرعت واقعی}) / (\text{نیرو - روز موجود})$$

سرعت واقعی جمع برآوردهای اولیه انجام شده در اسپرینت قبلی می باشد.

به این مثال توجه کنید: در اسپرینت قبلی 18 Story Point همراه 3 نفر لیزا - تام و سم طی 3 هفته انجام شد و جمع نیرو - روز اسپرینت 45 بود. و حالا ما می خواهیم توسط این اطلاعات سرعت اسپرینت بعدی را برآورد کنیم. برای پیچیده کردن مثال فرض کنید ، دیو به عنوان نیروی جدید به تیم اضافه شد. نیرو روز ما به 50 نیرو - روز افزایش می یابد.

محاسبه ضریب توجه اسپرینت قبلی :

$$45 \text{ نیرو-روز} / 18 \text{ Story Point} = 40\%$$

برآورد این اسپرینت :

$$40\% \times 50 \text{ نیرو-روز} = 20 \text{ Story Point}$$

خوب پس برآورد اسپرینت بعدی ما برابر با 20 Story Point شد. این بدین معنی می باشد که تیم تا زمانیکه جمع محتوای بک لاگ اسپرینت به حدود 20 برسد می تواند به داخل آن داستان اضافه نماید.

شروع اسپرینت



در این حالت تیم شاید فقط 4 داستان بالایی که جمعا 19 Story Point می شود را انتخاب کند یا برای پر شدن ظرفیت 5 داستان بالایی را انتخاب کند که در این حالت 24 Story Point خواهد شد.

فرض کنیم که تیم 4 داستان بالایی را انتخاب کرد، درست است که برآورد ما 20 بود ولی برآورد نهایی برای این اسپرینت در این حالت 19 SP خواهد بود.

تکنیک آب و هوای دیروز روشی سودمند است ولی همیشه درست عمل نمی کند. البته اگر فقط به یک اسپرینت قبلی نگاه کنیم. مثلا فرض کنید اسپرینت قبلی یک اسپرینت بد بوده است زیرا اکثر اعضای تیم مریض بودند و نمی توانستند کار کنند در این حالت ضریب توجه پایین خواهد بود و یا تیم در اسپرینت قبلی بیشتر در حال نصب ابزارها و تکنولوژی های جدید بوده است و در این حالت هم ضریب توجه پایین خواهد بود. در این تکنیک بهتر است برای اطمینان از عدد بدست آمده از معدل چندین اسپرینت قبلی استفاده شود.

اما اگر تیم یک تیم جدید باشد و هیچ پیشینه ای از آنها وجود نداشته باشد در آن مورد چه باید کرد؟ در این حالت می توان از پیشینه تیم های مشابه و هم شرایط آنها برای برآورد سرعت استفاده کرد.

اما اگر ما هیچ تیم دیگری نداشته باشیم که به پیشینه آنها نگاه کنیم چه باید بکنیم؟ ضریب توجه را حدس بزنید. خبر خوب این است که شما مجبورید فقط برای اسپرینت اول حدس بزنید و در اسپرینت های بعدی می توانید از آمار و اطلاعات اسپرینت های قبلی بهره ببرید.

ضریب توجهی که من به صورت پیش فرض برای تیم های جدید استفاده می کنم 70٪ می باشد. معمولا تیم ها در این حالت کارها را با کمی اضافه کار تمام می کنند.

از کدام روش برآورد سرعت استفاده می کنیم؟

من چندین روش برآورد سرعت ذکر کردم: برآورد توسط تجربه و احساس – محاسبه سرعت بر طبق آب و هوای دیروز و محاسبه سرعت بر اساس نیرو روزهای موجود و ضریب توجه.
اما از کدام روش استفاده می کنیم؟

ما معمولا از ترکیب اینها برای رسیدن به یک درجه اطمینان استفاده می کنیم. که واقعا زیاد طول نمی کشد.

ما به ضریب توجه و سرعت واقعی آخرین اسپرینت نگاه می کنیم. به مجموع منابع موجود این اسپرینت نگاه می کنیم و ضریب توجه آنها را محاسبه می کنیم. در مورد اختلاف بین این دو ضریب توجه بحث می کنیم و در صورت نیاز آن را همسان سازی می نماییم.

بعد از اینکه یک لیست از داستان ها برای اسپرینت انتخاب شد ، معمولا برای چک کردن از روش برآورد تجربه و احساس استفاده می کنیم. از تیم می خواهیم که اعداد را برای مدتی کنار بگذارند و فقط برای چند لحظه ای به امکان جایگیری داستان ها در اسپرینت فکر نمایند. اگر احساس کنیم که داستان بیش از حد بزرگ است آن را از لیست اسپرینت حذف می نماییم.

در پایان روز ، داستان های اسپرینت انتخاب شده اند و ضریب توجه ، منابع موجود و سرعت برآورد شده است و این بدین معنی است که ما به هدف جلسه برنامه ریزی اسپرینت دست یافته ایم.

چرا از کارت های ایندکس¹⁰ استفاده می کنیم؟

بیشتر زمان جلسه برنامه ریزی اسپرینت بر روی داستان های بک لاگ محصول صرف می شود. برآورد کردن آنها ، اولویت بندی دوباره آنها ، واضح سازی آنها ، شکستن آنها و غیره.

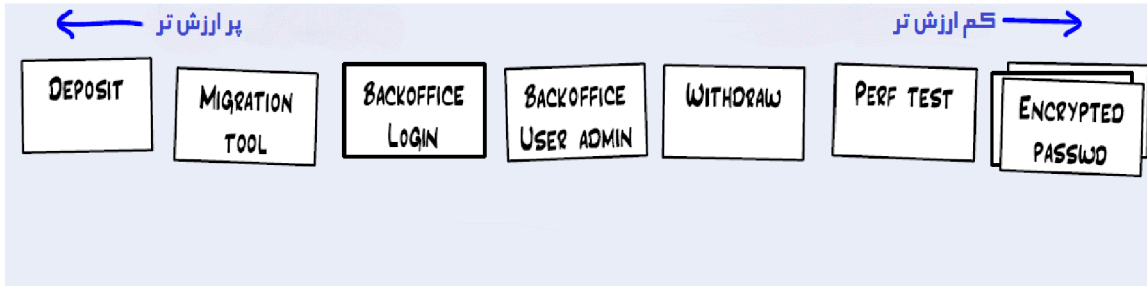
چگونه ما از آنها طی جلسه استفاده می کنیم؟

خوب ، در این حالت باید تیم از یک پروژکتور استفاده کند تا که فایل Excel بک لاگ محصول روی پرده نمایش ظاهر شود ، باید فردی هم مسئول صفحه کلید باشد تا بقیه را دعوت به بحث در مورد داستان بکند(معمولا یا مالک محصول و یا مدیر اسکرام). تیم درباره داستان ها بحث خواهد کرد و تغییرات را فرد صفحه کلید دار بر روی فایل Excel بک لاگ اعمال خواهد کرد.

¹⁰ Index Card

به نظر خوب می رسد؟ نه خیلی و معمولا درجا زدن است. اما تیم زمانی می فهمد که درجا زده است که زمان برنامه ریزی تمام شده است و هنوز هیچ یک از داستان ها در اسپرینت قرار نگرفته اند.

یک راه حل خوب ساخت کارت های ایندکس و قرار دادن آنها بر روی دیوار است (یا بر روی یک میز بزرگ).



این رابط کاربری بهتر از کامپیوتر و یا پروژکتور می باشد زیرا:

- افراد به دور این می ایستند و یا قدم می زنند => باعث به وجود آمدن هوشیاری و همیاری بیشتری می شود.
- همه احساس می کنند که درگیر کار هستند (و نه فقط کسی که دارای صفحه کلید است).
- چندین داستان به صورت همزمان قابل ویرایش خواهند بود.
- رتبه بندی دوباره آن ها ساده است – فقط کافی است کارت ها جابجا شوند.
- بعد از این جلسه می توان از همان کارت ها در اتاق تیم برای به وجود آوردن تابلو وظایف¹¹ استفاده کرد. (در بخش "چگونه بک لاگ اسپرینت را انجام می دهیم" بیشتر آشنا خواهید شد).

شما می توانید به صورت دستی داستان ها را بر روی کارت های ایندکس بنویسید و یا از یک اسکرپت برای تولید و چاپ ایندکس کارت های استفاده کنید.

¹¹ Task Board

Backlog item #55		اهمیت
سپرده گذاری		30
توضیحات		برآورد
به دیاگرام توالی UML نیاز است . نیازی به استفاده از encryption نیست .		
چگونگی دمو		
لاگین - باز کردن صفحه سپرده گذاری - سپرده گذاری مبلغ 10 دلار - بازکردن صفحه موجودی که باید موجودی 10 دلار اضافه شده باشد		

پ. ن: اسکریپتی برای این منظور در وبلاگ من^{۱۲} موجود است.

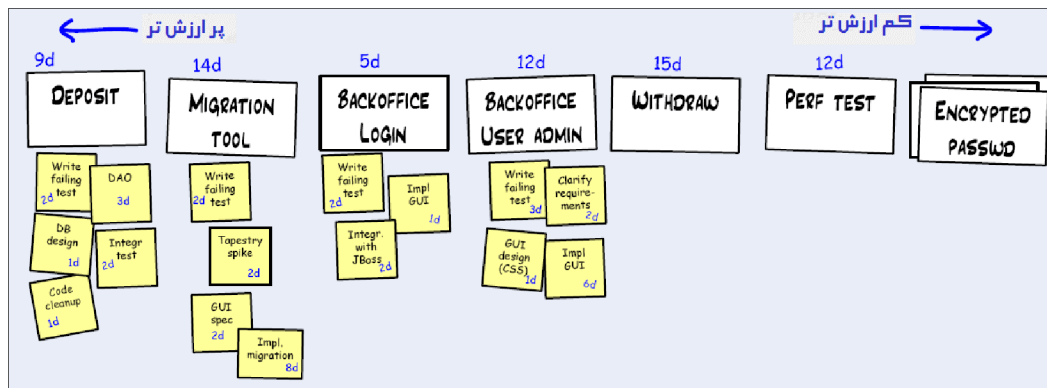
مدیر اسکرام ما بعد از جلسه برنامه ریزی تمام تغییرات موجود در کارت های ایندکس را بر روی فایل Excel به صورت دستی اعمال می نماید و این برای به روز نگه داشتن نسخه الکترونیک فایل بک لاگ می باشد.

یک نکته در مورد فیلد اهمیت کارت های ایندکس: از این فیلد ما در نسخه Excel برای مرتب سازی داستان ها استفاده می کردیم. اما در نسخه فیزیکی این فیلد زیاد به درد خور نخواهد بود و لازم نیست با هر بار جابجا کردن کارت ها این عدد را تغییر دهید. فقط دقت نمایید که بعد از مرتب سازی و اولویت دهی فیزیکی این عدد را برای نسخه Excel ویرایش کنید.

در این حالت برآورد زمانی راحت تر خواهد بود زیرا ما با شکست دادن داستان ها به وظایف خواهیم توانست که به برآورد زمانی معقول دست یابیم.

برای شکست دادن آنها می توانید از کاغذهای یادداشت کوچک رنگی استفاده کنید و وظیفه (فعالیت) مورد نظر را بر روی آنها یادداشت و زیر داستان مورد نظر بچسبانید.

¹² <http://blog.crisp.se/henrikkniberg>



ما فایل Excel بک لاگ محصول را با توجه به شکست داستان بر وظایف به دلیل ویرایش نمی کنیم:

- معمولاً یادداشت های وظایف اکثر طی اسپرینت در حال تغییرات خواهند بود و اکثراً ثابت نیستند و ثبت تغییرات دائم آنها در فایل Excel نیز غیر ضروری به نظر می رسد.
- مالک محصول نیاز نیست که در این سطح از جزئیات ورود پیدا کند.

همانند کارت های ایندکس، یادداشت های وظایف نیز می توانند در اتاق تیم بر روی تابلوی وظایف استفاده شوند.

تعریف Done

بسیار مهم است که تیم و مالک محصول بر روی تعریف واضحی از Done یا تکمیل شدن داستان ها توافق نظر داشته باشند. آیا زمانی است که کد به مخزن سورس ها فرستاده می شود؟ آیا داستان

زمانی تکمیل است که آن به قسمت تست ارسال شده باشد و تست یکپارچه^{۱۲} بر روی آن انجام شده باشد؟ ما از هر تعریف Done می‌توانیم استفاده کنیم: شاید زمانی که کدی تست می‌شود Done شود و یا شاید زمانیکه برنامه بر روی سرور راه اندازی می‌شود Done محسوب شود.

قبلا ما از چک لیست برای این منظور استفاده می‌کردیم ولی اکنون بدین گونه عمل می‌کنیم: "زمانی داستانی Done محسوب می‌شود که آزمایشگرها بگویند". Done برنامه نویس ها زمانی است که کدها را تحویل آزمایشگر بدهند و Done کلی داستان زمانی خواهد بود که داستان مورد نظر درخواست های مالک محصول را به درستی پوشش دهد که بررسی آن با آزمایشگر های ما می‌باشد.

ما به این حقیقت رسیدیم که همه داستان ها از طریق چک لیست قابل بررسی نیستند. مثلا با داستان "فرم کوئری کاربران" باید بسیار متفاوت تر از "دستور العمل عملیات" برخورد شود. به عبارتی دیگر تعریف Done می‌تواند به این سادگی باشد "پذیرفته شدن توسط تیم عملیاتی". به همین دلیل این تعریف بسیار بهتر از چک لیست ها به نظر می‌رسد.

اگر توضیحات ارائه شده باعث سردرگمی شما شد، من پیشنهاد می‌کنم که برای هر داستان به صورت جداگانه تعریفی از Done داشته باشید.

برآورد زمانی توسط Planning Poker

برآورد زمانی یکی از وظایف تیم می‌باشد و باید در این فرآیند تمام اعضای تیم حضور داشته باشند: چرا؟

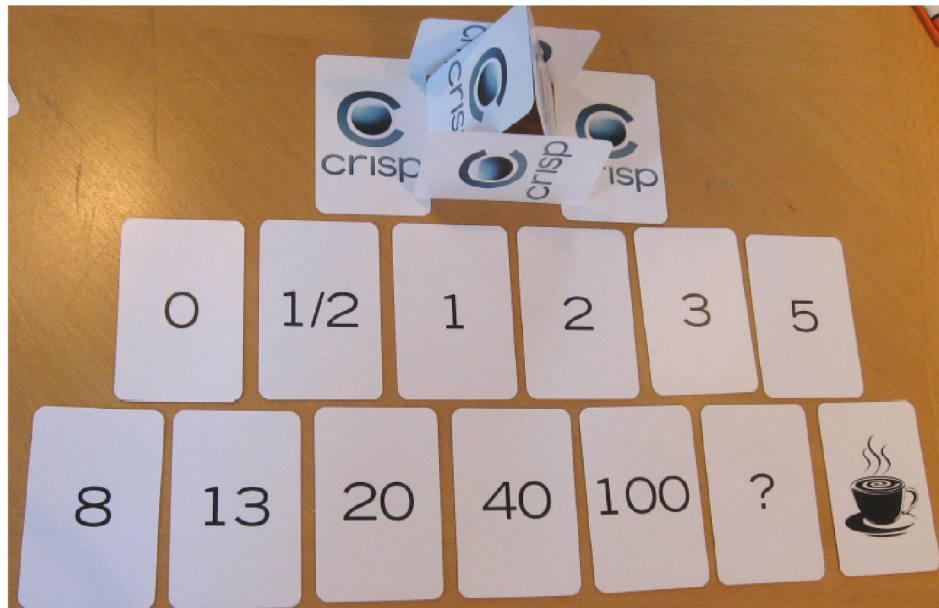
- در زمان برنامه ریزی ما دقیقا نمی‌دانیم که چه کسی قرار است چه بخشی از داستان ها را کار کند.
- داستان ها معمولا شامل چندین بخش و نیازمند چندین تخصص می‌باشند. (طراح رابط کاربری - کد نویسی - آزمایشگر و غیره)

¹³ Integration Test

- در زمان برنامه ریزی سولاتی برای وضوح بیشتر داستان پرسیده می شود. که این باعث بالا رفتن دانش تیم نسبت به داستان مورد نظر خواهد شد. در صورت وجود همه اعضا , سولات بهتری امکان مطرح شدن را خواهند داشت.
- وجود همه اعضا در برنامه ریزی باعث به وجود آمدن نقطه نظرات متفاوت خواهد شد که برای ایجاد یک برآورد خوب نیاز می باشد.

اگر از تیم بخواهید که بر روی داستانی برآوردی را انجام دهند , شخصی که بهتر از دیگران فهمیده است زودتر از دیگران جواب خواهد داد. متاسفانه این اولین جواب بر روی برآوردهای دیگران تاثیر گذار خواهد بود .

یک روش عالی برای جلوگیری از این مورد با نام **Planning Poker** وجود دارد (که به نظرم ابداع مایک کان^{۱۴} می باشد).



هر یک از اعضای تیم یک بسته کارت 13 تایی مانند شکل بالا دریافت می کنند. زمانیکه داستانی برآورد می شود یکی از کارت ها را به عنوان برآورد خود نشان می دهند(بر اساس Story Point). زمانیکه برآورد مورد نظر تکمیل و تایید شد کارت ها را دوباره جمع می کنند و دوباره برآورد بعدی. در این روش نظر و برآورد هیچ کس بر روی دیگران تاثیر نخواهد داشت.

¹⁴ Mike Cohn

اگر اختلاف بین برآورد ها خیلی زیاد باشد تیم سعی می کند که یک تصویر مشترک بین خود با شکست دادن داستان به وجود آورد. بعد دوباره برآورد انجام خواهد شد تا تقریباً برآورد ها نزدیک هم شوند.

به اعضای تیم گوش زد کنید که منظور r برآورد کل کار می باشد و نه فقط بخش مربوط به آنها. مثلاً نباید آزمایشگر فقط بخش مربوط به تست داستان را برآورد کند.

نکته قابل ملاحظه این می باشد که این اعداد یک سری عدد خطی و پیوسته نیستند. مثلاً مابین عدد 40 و 100 عدد دیگری وجود ندارد. چرا؟

این برای پرهیز از حس اشتباه دقیق کردن برآورد های بزرگ می باشد. اگر یک داستان به طور حدودی 20 Story Point برآورد گردید این دیگر صحیح نیست بحث کنیم که این باید 21 یا 19 شود ما به طور حدودی 20 را انتخاب کردیم. همه ما می دانیم که این یک داستان بزرگ می باشد و به همین دلیل برآورد آن مشکل می باشد و عدد 20 برآورد حدودی و تقریبی ما می باشد و نیاز به یک برآورد کاملاً دقیق در داستان های بزرگ نیست.

اگر واقعا می خواهید داستان های بزرگ را دقیق برآورد کنید r آن ها را به داستان های کوچک تر بشکنید و به جای برآورد یک داستان بزرگ آن ها را برآورد نمایید.

و به خاطر داشته باشید که شما نمی توانید در این برآورد تقلب نمایید - مثلاً عدد 2 را همراه 5 به منظور 7 Story Point نشان دهید. عدد 7 وجود ندارد یا باید 5 و یا باید 8 را انتخاب کنید.

توضیح بعضی از کارت های ویژه:

0 = "این داستان قبلاً انجام شده است" یا "انجام این داستان چند دقیقه بیش تر طول نخواهد کشید".

؟ = "من واقعا در مورد این داستان هیچ نظری ندارم".

فنجان قهوه = "من واقعا خسته شدم. کمی استراحت خوب است".

واضح سازی داستان ها

بدترین حالت زمانی اتفاق می افتد که داستانی به طور کامل تکمیل می شود و در زمان ارائه Demo اسپرینت ، مالک محصول می گوید: "این خیلی خوب است ولی این چیزی نیست که من خواسته بودم!".

چگونه مطمئن می شوید که درک مالک محصول از یک داستان همانند درک تیم از آن داستان می باشد؟ یا تمام اعضای تیم یک برداشت و درک را از همان داستان دارند؟ خوب ، شما نمی توانید چنین اطمینانی بدست آورید. اما تکنیک های ساده ای برای مشخص کردن کج فهمی های بزرگ موجود می باشد. ساده ترین تکنیک این است که مطمئن شوید تمام فیلدها برای تمام داستان پر شده باشند.

مثال 1 :

تیم و مالک محصول خوشحال هستند که برنامه ریزی اسپرینت را انجام داده اند و حاضرند که جلسه را تمام کنند. مدیر اسکرام می گوید: "یک لحظه صبر کنید ، این داستان با نام 'اضافه کردن کاربر' برآورد زمانی ندارد ، بیایید این را برآورد نمایم." بعد از چند دور planning poker تیم برآورد 20 Story Point را برای داستان بر می گزیند. در این هنگام مالک محصول با خشم به پا می خیزد و می گوید "20؟! ". بعد از چند دقیقه بحث گرم ، تیم متوجه می شود که درک دامنه داستان دچار اشتباه شده است. تیم فرض کرده بود که منظور از اضافه کردن کاربر یک رابط کاربری پیشرفته تحت وب همراه با امکانات حذف و ویرایش کاربر خواهد بود در حالیکه منظور مالک محصول "اضافه کردن کاربر توسط کوئری SQL در سمت پایگاه داده" بود. در برآورد دوباره تیم در 5 Story Point به توافق رسید.

مثال 2 :

تیم و مالک محصول خوشحال هستند که برنامه ریزی اسپرینت را انجام داده اند و حاضرند که جلسه را تمام کنند. مدیر اسکرام می گوید: "یک لحظه صبر کنید ، برای داستان با نام 'اضافه کردن کاربر' ، دمای آن چگونه خواهد بود؟". بعد از کمی مکث یکی از اعضا بر می خیزد و می گوید: "ما باید نخست وارد سایت شویم و در آنجا لاگین نمایم و سپس ... ". مالک محصول صحبت او را قطع می نماید: "لاگین در وب سایت؟ نه نه نه! این قسمتی از وب سایت نخواهد بود بلکه یک امکان برای مدیران فنی پایگاه داده می باشد".

توضیحات "چگونگی دمو" باید بسیار خلاصه باشند زیرا اگر اینگونه نباشد نمی توانید جلسه برنامه ریزی را به موقع تمام نمایید. می توانید از قالب "این را انجام بده - آنگاه آنرا - سپس آن را بررسی کن" برای خلاصه کردن توضیحات استفاده کنید.

من این را متوجه شده ام که این توضیحات خلاصه و مختصر جلوی کج فهمی های بزرگی را در سطح دامنه و خود داستان گرفته اند.

شکستن داستان های بزرگ به داستان های کوچک تر

داستان ها نباید زیاد کوچک و نه زیاد بزرگ باشند (در اصلاح برآورد). انجام دادن یک داستان 40 Story Point ای به صورت نیمه تکمیل ریسک بزرگی خواهد بود. فرض کنید دو داستان اولیه بک لاگ هر دو 40 Story Point می باشند و ظرفیت اسپرینت ما 70 پوینت می باشد. شما به همین خاطر فقط یکی از داستان های را انتخاب خواهید کرد. ولی اگر داستان 7 بعد از اسپرینت قابل تحویل نباشد 7 کل اسپرینت بدون نتیجه خواهد ماند.

پس بهترین راه در این حالت شکست داستان های بزرگ به داستان های کوچکتر می باشد. اما مطمئن شوید که طی یک داستان کوچک ارزش تجاری تحویل مالک محصول خواهد شد. به این مورد اصلاحا قابل ارائه¹⁵ با ارزش تجاری¹⁶ می گویند.

تلاش ما برای ایجاد داستان هایی با وزن 2-8 نیرو-روز می باشد. معمولا سرعت ما در حدود 40-60 برای یک تیم می باشد که معمولا می توانند 10 داستان را در هر اسپرینت انجام دهند. بعضی مواقع هم 5 و بعضی مواقع هم 15 داستان .

شکست دادن داستان ها به وظایف

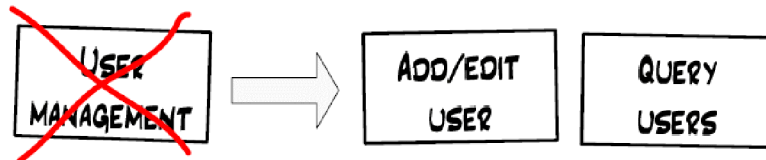
یک لحظه صبر کنید. چه فرقی مابین یک داستان و یک وظیفه (Task) وجود دارد؟ سوال کاملا به جایی است.

¹⁵ Deliverables

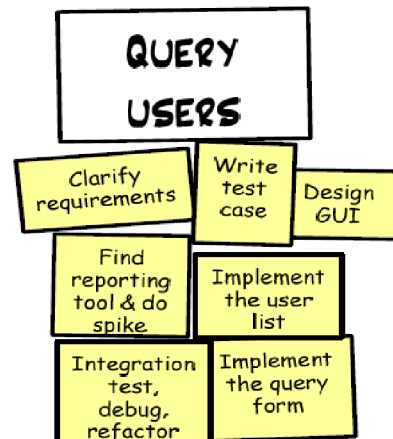
¹⁶ Business Value

تفاوت کاملاً واضح می باشد. داستان‌ها چیزهای قابل ارائه‌ای هستند که مالک محصول مواظب و مراقب آنها هست. وظایف و یا Task‌ها چیزهای غیر قابل ارائه‌ای هستند که تحت مراقبت مالک محصول نیستند.

نمونه‌ای از شکست داستان به داستان‌های کوچکتر:



نمونه‌ای از شکست داستان به Task‌ها:



نکات مهم در این باب:

- تیم‌های جدید اسکرام جهت صرف کردن زمان برای شکستن داستان به وظایف بی‌میل هستند و احساس می‌کنند این کار عملی مانند متدولوژی آبشاری می‌باشد که ما می‌خواهیم تمام وظایف یک داستان را استخراج کنیم.
- برای فهم بهتر و واضح شدن داستان، شکست کامل داستان به وظایف حتی مثل متد آبشاری مفید هست.

- اینگونه شکستن داستان ها به وظایف باعث حفظ زمان و جلوگیری از برخی کارهای اضافی در آینده خواهد شد و باعث به وجود آمدن یک برنامه ریزی واقعی تر نیز خواهد بود.
- اینگونه شکست دادن داستان ها به وظایف باعث پرمفعت تر شدن جلسات اسکرام روزانه نیز خواهد شد (به بخش "چگونه اسکرام روزانه را انجام می دهیم" مراجعه کنید).

نکته – ما از روش TDD^{۱۷} یا توسعه آزمایش محور استفاده می کنیم که معنی آن این است که اولین کار برای پیاده سازی داستان "نوشتن Failing تست ها" می باشد و آخرین کار "رفاکتور" (= بهبود خوانایی کد و حذف کدهای تکراری) می باشد.

تعریف زمان و مکان برای جلسه روزانه اسکرام

یکی از پر فراموش ترین اهداف جلسه برنامه ریزی اسپرینت، تعیین زمان و مکان جلسه روزانه اسکرام می باشد. بدون این اسپرینت شما شروع خوبی نخواهد داشت. اولین جلسه اسکرام روزانه برای اینکه هر کس تصمیم بگیرد که کجا کار کند ضروری است.

من جلسات صبح را بیشتر می پسندم. ولی می توان این جلسات را در نیمه روز و یا بعد از ظهر نیز برگزار کرد.

معایب جلسات اسکرام در بعد از ظهر: زمانی که صبح به سر کار می آیید مجبور خواهید بود کارهایی را که دیروز به افراد گفته اید که می خواهید امروز انجام بدهید را به یادتان بیاورید.

معایب جلسات اسکرام در صبح: زمانی که صبح به سر کار می آیید مجبور خواهید بود کارهایی که دیروز انجام داده اید را به یادتان بیاورید تا به بتوانید به دیگران گزارش دهید.

به نظر من این مورد اشتباه می باشد. زیرا در این جلسات مهم نیست که شما چه چیزی را انجام داده اید، بلکه این مهم است که شما امروز می خواهید چه چیزی را انجام بدهید.

¹⁷ Test Driven Development

رویه پیش فرض ما این است که زودترین زمان ممکن که هیچ کس غر نزند را برای این جلسه انتخاب می نماییم. معمولاً 9:00 , 9:30 یا 10:00 . مهمترین مورد این است که همه افراد شرکت کننده با تمام وجود این زمان را قبول داشته باشند.

کجا خط بکشم؟

بسیار خوب ، زمان اتمام جلسه فرا رسیده است. از میان کارهایی که می خواهیم در جلسه برنامه ریزی اسپرینت انجام دهیم ، چه چیزهایی را می توانم در صورت اتمام زمان جلسه کنار بگذارم و یا خط بزنم؟

خوب من از لیست اولویت زیر استفاده می کنم :

اولویت 1 : هدف اسپرینت و زمان ارائه دمو. اینها حداقل هائی هستند که برای شروع اسپرینت نیاز هست. تیم یک هدف و یک زمان اتمام کار دارد پس می توان برای رسیدن به هدف از کارهای موجود در بک لاگ محصول استفاده کند. آیا این خیلی بد نیست ؟ بلی. شما باید فردا یک جلسه جدید برنامه ریزی اسپرینت برگزار کنید. من هیچ وقت اسپرینت را با اطلاعات خیلی کم شروع نمی کنم.

الویت 2 : لیست کارهایی که تیم برای انجام دادن در اسپرینت قبول کرده است.

الویت 3 : پر بودن فیلد برآورد برای داستان های موجود در اسپرینت.

الویت 4 : پر بودن فیلد "چگونگی دمو" برای داستان های موجود در اسپرینت.

الویت 5 : محاسبه سرعت و منابع موجود به صورت واقعی برای چک کردن برنامه ریزی اسپرینت. شامل لیست اعضای تیم و تعهدات (بدون این نمی توان سرعت را محاسبه کرد).

الویت 6: تعیین زمان و مکان جلسه روزانه اسکرام. تصمیم این کار شاید فقط چند دقیقه طول بکشد ولی اگر زمان تمام شد، مدیر اسکرام می تواند بعداً از طریق ایمیل و یا ... همه را از زمان و مکان جلسه با خبر کند.

الویت 7: شکستن داستان ها به وظایف. این کار علاوه بر این جلسه می تواند در جلسات روزانه اسکرام نیز انجام شود ولی برای بر هم نزدن اهداف آن جلسه بهتر است شکستن داستان ها به وظایف در همین جلسه انجام شود.

داستان های فنی

در این جا یک مسئله اساسی وجود دارد با نام داستان های فنی یا داستان غیر عملیاتی و یا هر چیز دیگری که شما می نامید.

این آیتم ها شامل مواردی می شوند که باید انجام شوند ولی قابل ارائه نیستند، به طور مستقیم ربطی به هیچ داستانی ندارند و ارزش تجاری مستقیم برای مالک محصول به وجود نمی آورند.

ما به این موارد داستان های فنی¹⁸ می گوییم.

برای مثال:

- نصب سرور Build مداوم¹⁹
 - چرا باید این داستان انجام شود: به خاطر اینکه زمان زیادی برای توسعه گران حفظ می کند و از ریسک بروز انفجار-بزرگ در هنگام یکپارچه سازی اسپرینت می کاهد.
 - نوشتن خلاصه طراحی سیستم
 - چرا باید این داستان انجام شود: به خاطر اینکه توسعه گران طی مدتی سراسر طراحی سیستم را از یاد می برند و شروع به کد نویسی متناقض با طراحی سیستم می کنند. داشتن یک سند "عکس بزرگ" که جلوی این تناقضات را بگیرد مفید خواهد بود.
- رفاکتور لایه DAO

¹⁸ Tech Stories

¹⁹ Continuous build server

○ چرا باید این داستان انجام شود: به خاطر اینکه این لایه به اندازه کافی شلوغ و درهم می باشد و بیشتر باگ ها از این لایه حادث می شوند. با رفاکتور کردن کدهای داخل آن می توانیم هزینه باگ زدایی و نگه داری سیستم را بسیار کاهش دهیم.

• به روز کردن Jira (سیستم پیگیری باگ)

○ چرا باید این داستان انجام شود: این نسخه از این برنامه بسیار پر باگ و وقت گیر هست. با به روز کردن آن می توانیم در وقت بسیار صرفه جویی کنیم.

آیا این داستان ها با داستان های دیگر به صورت مستقیم در ارتباط می باشند؟ چه کسی باید این داستان ها را اولویت بندی کند؟ آیا مالک محصول هم باید در این داستان ها مداخله داشته باشد؟

ما سعی می کنیم در برخورد با داستان های فنی ، آن ها را به صورت داستان های درجه 1 در نظر بگیریم. اما زمانی که آنها در داخل بک لاگ محصول قرار می گیرند همانند سبب ها و پرتقال ها داخل یک سبد می باشند. سبب ها در بالا و پرتقال ها پایین. بدلالی مالک محصول دوست دارد داستان های فنی در اولویت پایین قرار گیرند: "دوستان ، من مطمئن هستم که ایجا یک سرور Build متوالی کار بسیار پر اهمیتی هست ، ولی بیاید آن داستان که پر اهمیت تر و ارزش تجاری دارد را ایجاد کنیم بعد از آن شما می توانید داستان های فنی خود را ایجاد کنید".

در بعضی از مواقع حق با مالک محصول است ولی نه همیشه. ما به این نتیجه دست یافته ایم که همیشه مالک محصول استحقاق اظهار نظر در مورد داستان های فنی را ندارد. این ها مواردی هستند که ما انجام می دهیم:

1. سعی برای پرهیز از داستان های فنی. همیشه سعی می کنیم که داستان های فنی را در داخل داستان های معمولی در داخل بک لاگ محصول بگنجانیم تا مالک محصول در برخورد با آنها به مشکل برخورد.

2. اگر ما نتوانیم داستان فنی را به یک داستان معمولی تغییر دهیم ، سعی می کنیم آن داستان فنی را در قالب یک وظیفه (Task) در خلال داستان دیگری انجام دهیم. برای مثال داستان فنی "رفاکتور لایه DAO" را می توان در خلال داستان "ویرایش کاربر" که شامل لایه DAO هم می شود انجام داد.

3. اگر هر دو روش بالا امکان پذیر نبودند، داستان را در قالب داستان فنی تعریف کنید و آن ها را در لیست مجزایی به نام لیست داستان های فنی قرار دهید. به مالک محصول اجازه مشاهده آنها را بدهید ولی حق ویرایش را نه. برای آنها از پارامتر "ضریب توجه" و "سرعت برآورد شده" جهت مذاکره با مالک محصول استفاده کنید و زمان هایی را در اسپرینت برای پیاده سازی آن داستان ها به توافق برسید.

برای مثال :

تیم : "ما باید تعدادی داستان فنی در خلال این اسپرینت انجام دهیم. به همین خاطر ما در نظر داریم 10٪ از این اسپرینت را به این موضوع اختصاص دهیم. پس ضریب توجه ما از 75٪ به 65٪ درصد کاهش خواهد یافت."

مالک محصول : "امکان ندارد! ما وقت کافی نداریم."

تیم : "خوب، فقط کافی است به اسپرینت قبلی نگاه کنید. ما در اسپرینت قبلی SP 80 برآورد کرده بودیم که فقط SP 30 انجام شد."

مالک محصول : "من نمی دانم، ما برای جبران کم کاری اسپرینت قبلی به این نیاز داریم که کارهای اضافی زیادی انجام بدهیم."

تیم : "دلیل اینکه سرعت ما در آن اسپرینت بسیار پایین آمده بود این بود که ما زمان زیادی برای کنار هم قرار دادن و همساز سازی قسمت های مختلف و آزمایش آنها صرف کردیم."

مالک محصول : "خوب؟"

تیم : "خوب اگر کارهای انجام شده در آن اسپرینت را تکرار نماییم سرعت ما در این اسپرینت افزایش خواهد یافت."

مالک محصول : "خوب؟"

تیم : "خوب ما قصد داریم 10٪ از زمان این اسپرینت را برای نصب سرور Build متوالی و دیگر داستان های فنی صرف نماییم تا بتوانیم برای همیشه و تمام اسپرینت ها زمان یکپارچه سازی و دیگر مسائل مربوطه را حل نماییم."

مالک محصول : "واقعا؟ چرا ما این ها را قبلا انجام ندادیم؟"

تیم : "به خاطر اینکه شما از ما نخواستید بودید ..."

مالک محصول : "بسیار خوب، به نظر خوب می رسد. همین کار را می کنیم."

مهم ترین موارد در این گونه مسائل این هست که هوشیاری مالک محصول را در مورد اهمیت این مسائل روشن کنید. و به او اجازه بدهید تا به اهمیت قضایا و پشت پرده فنی پی ببرد.

اگر مالک محصول شما هم مانند مالک محصول ما فرد لایق و مسئولی بود تا حد امکان اجازه بدهید تا از کم و کیف داستان ها (فنی ها) با خبر و با اطلاع باشد تا بتواند اولویت بندی موثری انجام دهد. مگر شفافیت و وضوح یکی از ارزش های اصلی اسکرام نیست؟

سیستم پیگیری باگ در مقابل بک لاگ محصول

Excel از قالب خوبی برای نگه داری داستان های بک لاگ برخوردار می باشد. اما هنوز شما علاوه بر این نیاز به یک سیستم پیگیری باگ دارید که Excel مناسب این کار نیست. ما از Jira برای این کار استفاده می کنیم.

چگونه موارد و آیتم های درون Jira به جلسه برنامه ریزی اسپرینت آورده می شوند؟ منظور من این است که ما نمی توانیم باگ ها را در نظر نگیریم و فقط بر روی داستان ها تمرکز کنیم .

ما از روش های مختلفی استفاده می نمایم :

1. مالک محصول از مهمترین باگ های درون Jira یک پرینت می گیرد و به جلسه برنامه ریزی اسپرینت می آورد و آنها را در کنار دیگر داستان بر روی دیوار قرار می دهد .
2. مالک محصول داستان هایی که به آیتم های Jira اشاره داشته باشد ایجاد می کند . برای مثال : "رفع مهمترین باگ های دفتر پستی با شماره های Jira-124, Jira-126 و Jira-180".
3. رفع خطا و باگ ها به کارهای بعد اسپرینت واگذار می شود. مثلا تیم ضریب توجه خود را پایین در نظر می گیرد (مثلا 50٪) در این حالت زمان خالی در آخر اسپرینت برای رفع خطاها به وجود می آید.
4. قرار دادن بک لاگ محصول در داخل Jira . در این حالت باگ ها همانند داستان های معمولی در نظر گرفته می شوند.

برای ما تمام استراتژی های بالا با ارزش هستند و در واقع نمی توان گفت کدامیک بهترین است. از تیم به تیم و از اسپرینت به اسپرینت روش های مختلفی جوابگو بوده اند. ولی من به شخصه اولین روش را به خاطر سادگی آن بیشتر می پسندم.

جلسه برنامه ریزی اسپرینت بالاخره تمام شد

من هیچ وقت فکر نمی کردم که فصل برنامه ریزی اسپرینت اینقدر طولانی بشود. حدس می زنی بازتاب نظرات من در مورد برنامه ریزی اسپرینت مهمترین کاری باشد که شما در اسکرام انجام می دهید. زمان بیشتری را بر روی انجام دادن صحیح این موارد صرف کنید و بعد آن همه چیز آسان خواهد شد.

زمانی جلسه برنامه ریزی اسپرینت موفق می باشد: کسانی که از جلسه خارج می شوند با خنده خارج شوند. زمانی که فردای آن روز از خواب بر میخیزند با خنده بلند شوند و زمانی که اولین جلسه روزانه اسکرام را برگزار می کنند همراه با خنده باشد.

بعد این اگر همه چیز هم به بدترین وضع ممکن جلو رفت شما نمی توانید جلسه برنامه ریزی اسپرینت را مقصر بدانید.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

5

چگونگی ارتباطات در طی اسپرینت ها

این بسیار مهم است که کل سازمان و شرکت را از کارهایی که انجام می شود با خبر کنید. به عبارتی افراد دیگر و خارج از تیم می توانند فرض های اشتباه از کارهای در حال انجام داشته باشند که این ممکن است به وخیم شدن اوضاع منجر شود.

ما از "صفحه اطلاعات اسپرینت" استفاده می کنیم .

تیم کله خراب ها - اسپرینت 15

هدف اسپرینت

- ارائه نسخه بتا

بک لاگ اسپرینت (برآورد ها داخل پارانتز)

- سپرده گذاری (3)
- ابزار مهاجرت داده (8)
- لاگین (5)
- سیستم مدیریت کاربر (5)

سرعت برآورد شده : 21

زمانبندی

- مدت اسپرینت : 6-1-2009 تا 24-1-2009
- اسکرام روزانه : 9:30 - 9:45 - در اتاق تیم
- دموی اسپرینت : 24-1-2009 - در کافه تریا

تیم

- جیم
- اریکا (مدیر اسکرام)
- تام (75%)
- ایوا
- جان

بعضی مواقع توضیحاتی راجع به چگونگی دموی داستان ها نیز در این صفحه ذکر می کنیم. در اولین زمان ممکن بعد از جلسه برنامه ریزی اسپرینت ، مدیر اسکرام صفحه اطلاعات اسپرینت را ایجاد و بر روی ویکی قرار می دهد و علاوه بر این ایمیلی با این مضمون برای تمام سازمان ارسال می کند.

موضوع : 15 امین اسپرینت تیم کله خراب ها شروع شد

سلام به همه. تیم کله خراب ها 15 امین اسپرینت خود را با هدف ارائه نسخه بتا شروع کرد. زمان ارائه دمو این اسپرینت 24 نوامبر خواهد بود . برای مشاهده اطلاعات اسپرینت به صورت تفضیلی به لینک زیر مراجعه کنید :

<http://wiki.mycompany.com/kalekharab/sprint15>

ما همچنین یک صفحه داشبورد بر روی ویکی خود داریم که تمام اسپرینت های در حال انجام بر روی آن قابل مشاهده می باشند:

صفحه داشبورد

اسپرینت های در حال انجام

- تیم X اسپرینت 15
- تیم Y اسپرینت 12
- تیم Z اسپرینت 1

بعلاوه مدیر اسکرام از صفحه اطلاعات اسپرینت یک پرینت می گیرد و در بیرون اتاق تیم می چسباند. کسانی که از آنجا رد می شوند می توانند با نگاه به آن صفحه متوجه فعالیت تیم شوند.

زمانیکه به اواخر اسپرینت نزدیک می شویم ، مدیر اسکرام ایمیلی مبنی بر یادآوری ارائه زمان دمو برای همه ارسال می کند.

موضوع : دموی اسپرینت 15 تیم کله خراب ساعت 13:00 در کافه تریا

سلام به همه . ما بسیار خرسند می شویم که فردا در ساعت 13:00 واقع در کافه تریا جهت نمایش دموی اسپرینت 15 حضور داشته باشید . ما در این جلسه نسخه بتا را نمایش خواهیم داد .

برای مشاهده اطلاعات اسپرینت به لینک زیر مراجعه کنید :

<http://wiki.mycompany.com/kalekharab/sprint15>

ارسال این ایمیل باعث خواهد شد که هیچ کس بعدا بهانه نگیرد که من نمی دانستم کی و کجا قرار است دمو را نشان دهید.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

6

بک لاگ اسپرینت را چگونه انجام می دهیم

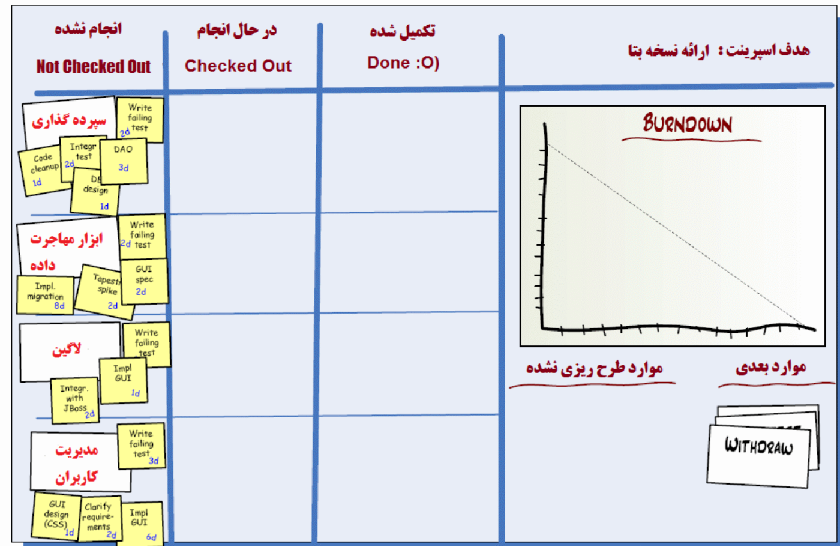
جلسه برنامه ریزی اسپرینت تمام شد و ما با افتخار به تمام جهان گفتیم که می خواهیم اسپرینت جدیدی را شروع کنیم. نوبت این رسیده است که مدیر اسکرام بک لاگ اسپرینت را ایجاد کند. این کار باید بعد از جلسه برنامه ریزی اسپرینت و قبل از اولین جلسه روزانه اسکرام باشد.

قالب بک لاگ اسپرینت

ما تجربه استفاده از فرمت های گوناگون بک لاگ اسپرینت را داریم. اعم از: Jira یا Excel و یا تابلوی وظایف فیزیکی. ما در ابتدا بیشتر از فرمت Excel برای بک لاگ اسپرینت استفاده می کردیم با استفاده از چارت های این برنامه نمودار Burndown را به صورت اتوماتیک تولید می کردیم و در کل خوب بود ولی در اینجا نمی خواهیم این مورد را شرح دهیم.

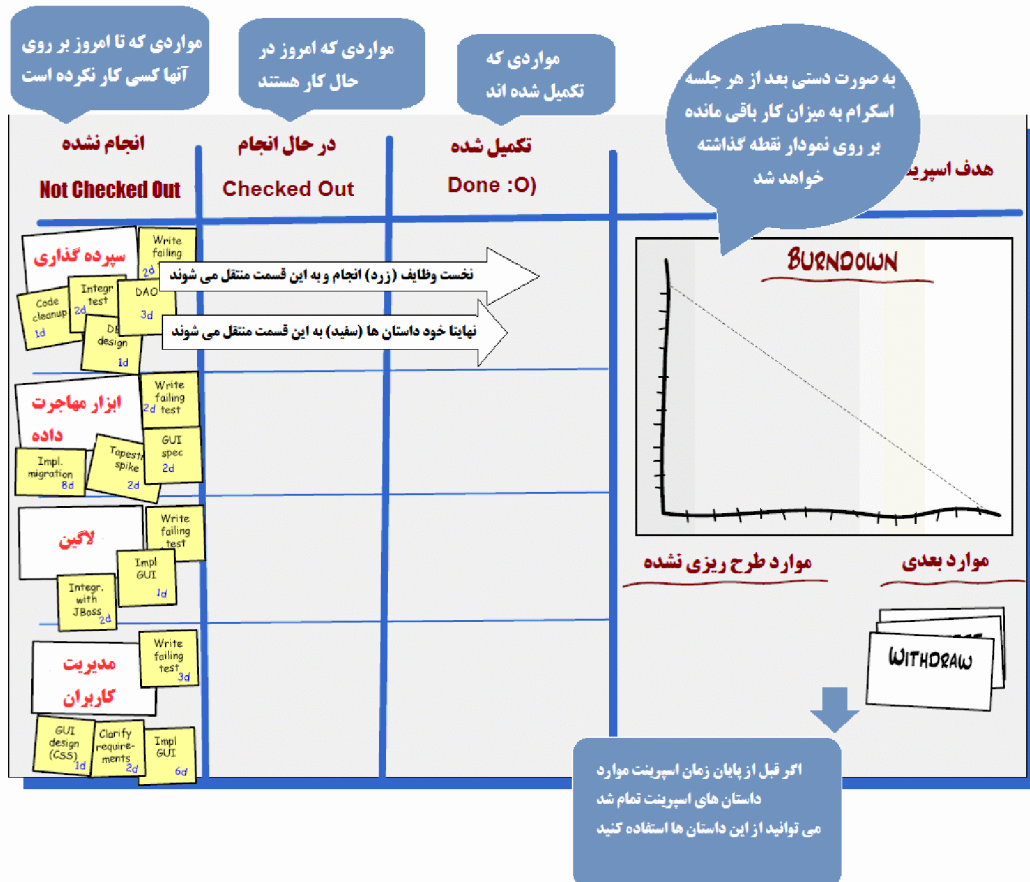
به جای این می خواهیم با پرداخت به جزئیات بهترین قالبی که ما تا به حال دیده ایم را شرح دهیم: بک لاگ اسپرینت در قالب تابلوی وظایف نصب شده بر روی دیوار.

یک دیوار بزرگ خالی در اتاق تیم پیدا کنید. دیوار را از موارد کم ارزش مانند لوگوی شرکت و یا نمودار های قدیمی پاک کنید. یک صفحه کاغذی بزرگ بر روی دیوار بچسبانید (حداقل 2x2 متر و یا 3x3 متر برای تیم های بزرگ). سپس صفحه را به مانند شکل زیر در آورید:



شما می توانید از وایت برد هم استفاده کنید. اما بدلیل اینکه از وایت برد بعدا استفاده خواهد شد پس بهتر است از دیوار های خالی که بی استفاده مانده اند استفاده کنید.

تابلوی وظایف چگونه کار می کند

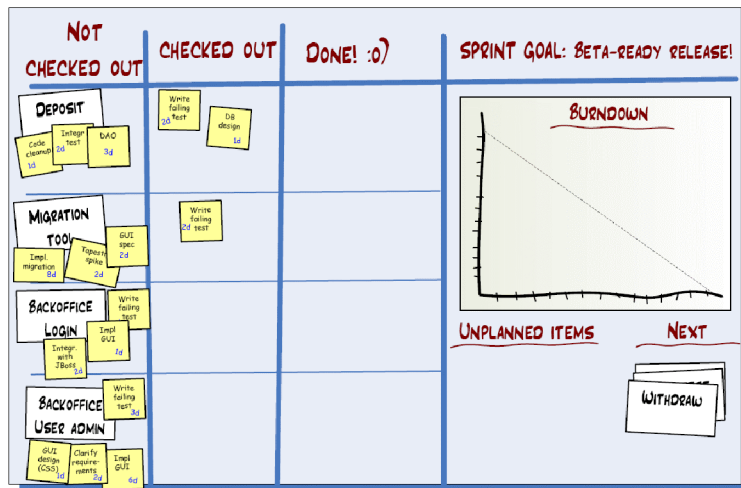


مسئله شما می‌توانید ستون‌های دیگری نیز اضافه کنید. برای مثال "در انتظار تست یکپارچه سازی" یا "کنسل شده‌ها" یا هر چیز دیگری. فقط قبل از اضافه کردن ستون دقیقاً توجه نمایید که این ستون واقعا مورد نیاز باشد؟

با توجه به تجربه‌های انجام شده من فقط از این سه ستون برای بالا بردن میزان ساده سازی کار استفاده می‌کنم و فقط زمانی از ستون‌های جدید استفاده می‌کنم که تداخلی با اصل سادگی نداشته باشند.

مثال 1 – بعد از اولین جلسه روزانه اسکرام

بعد از اولین جلسه روزانه اسکرام ممکن است که تابلوی وظایف بدین شکل باشد:

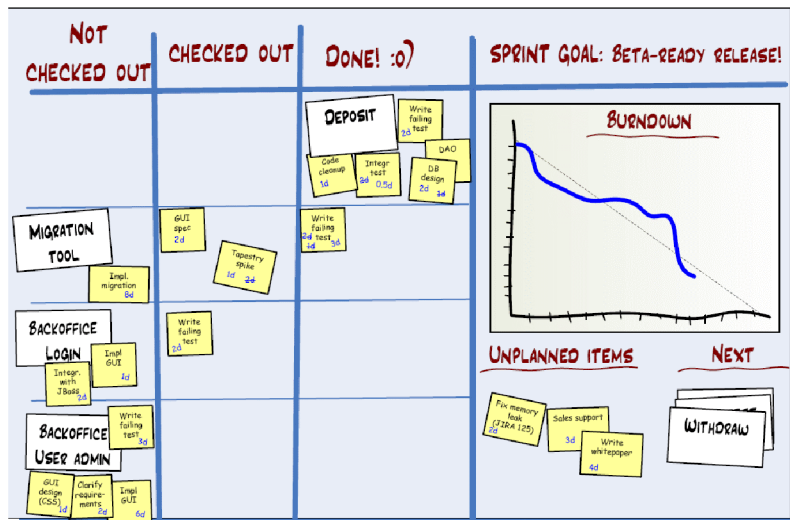


همان‌طور که مشاهده می‌فرمایید این 3 وظیفه امروز برای پیاده‌سازی انتخاب شده‌اند و امروز بر روی آنها توسط اعضای تیم کار انجام خواهد شد.

معمولاً در تیم‌های بزرگ برای ایجاد ارتباط بهتر، بر روی هر وظیفه در حال انجام یک لیبل که همان نام و یا عکس انجام‌دهنده است نصب می‌شود تا بقیه بدانند که آن داستان را چه کسی انجام می‌دهد.

مثال 2 – بعد از چند روز

بعد از چند روز ممکن است که تابلوی وظایف بدین شکل باشد:



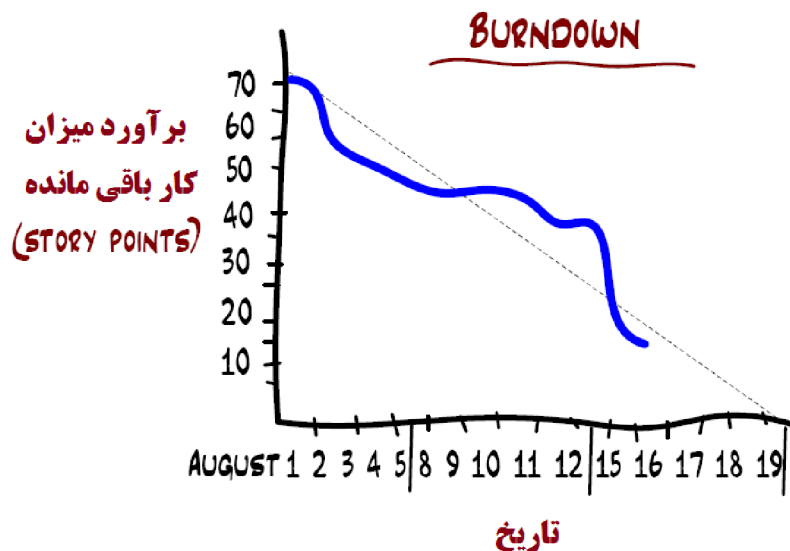
همانطور که مشاهده می کنید ما داستان سپرده گذاری (Deposit) را تکمیل کرده ایم و علاوه بر این یکی از وظایف دومین داستان را نیز تکمیل کرده ایم و چند وظیفه از داستان های دیگر هم در حال پیاده سازی می باشند.

شما می توانید 3 مورد (سمت راست پایین) طرح ریزی نشده را نیز در شکل بالا مشاهده کنید. این موارد می توانند در جلسه بازیابی عملکرد اسکرام مفید باشند. شکل پایین نمایی از یک بک لاگ اسپرینت واقعی در اواخر اسپرینت می باشد. توجه نمایید که باید در هر اسپرینت جدید بک لاگ اسپرینت را خالی و تمیز نمایید.



نمودار Burndown چگونه کار می کند

اجازه بدهید بر روی نمودار برن داوون دقت کنیم.



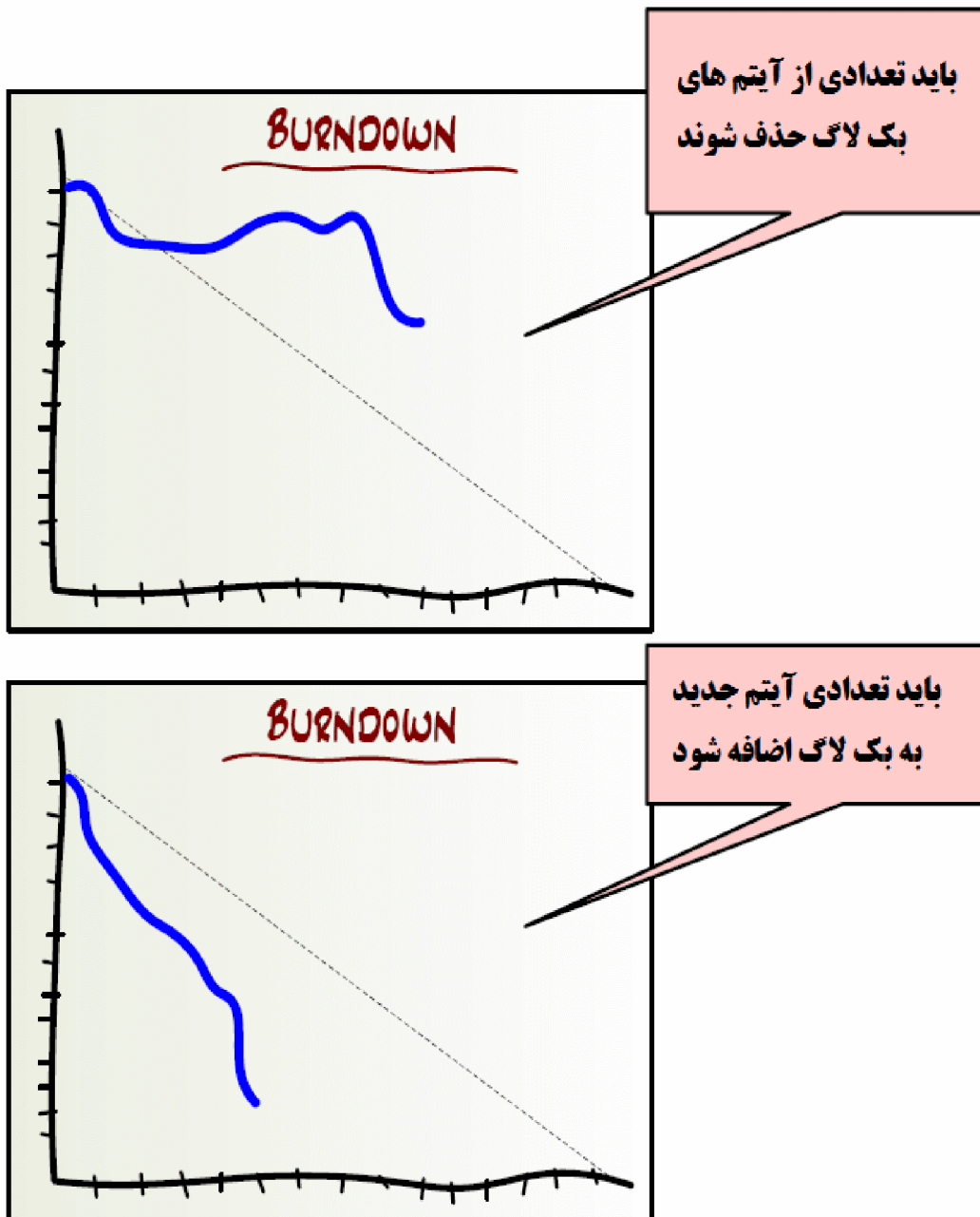
این نمودار این را نشان می دهد که :

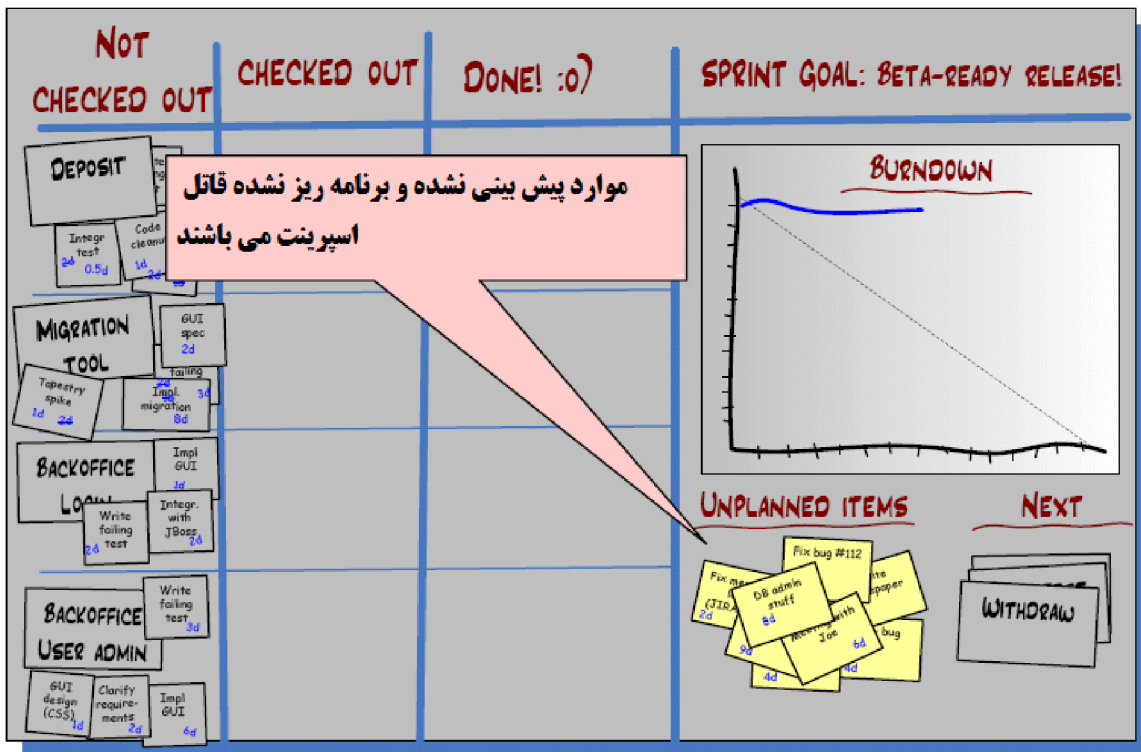
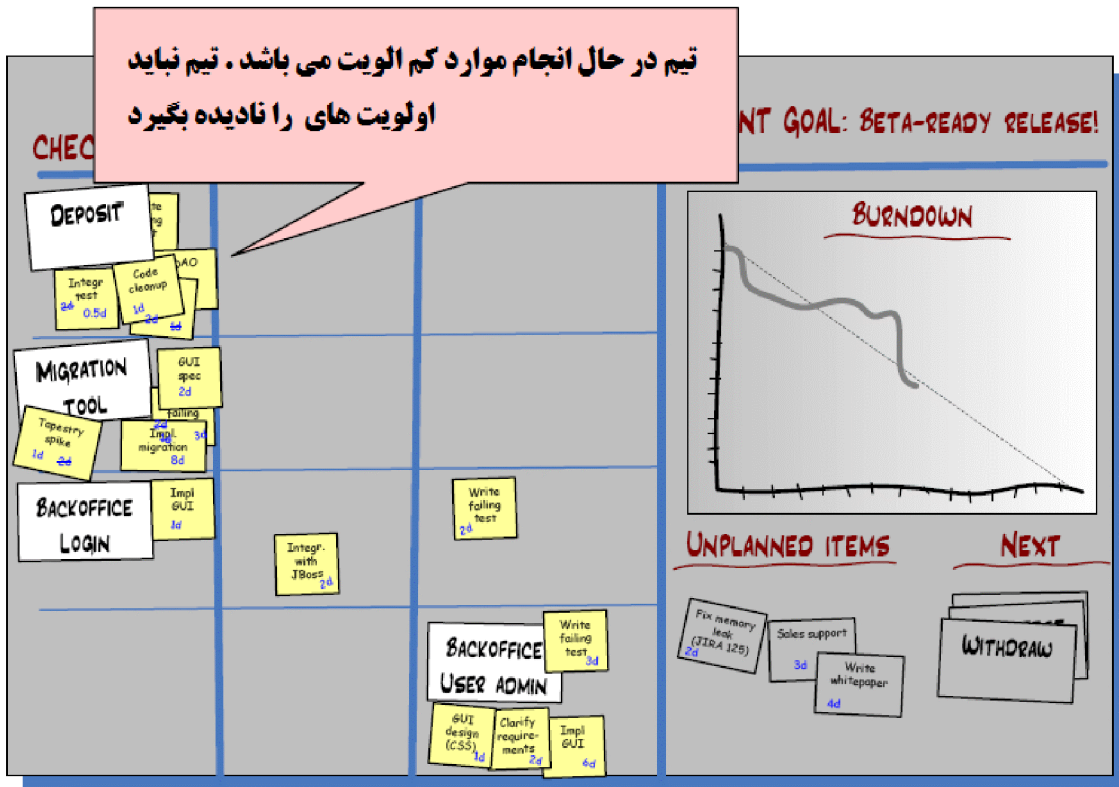
- در اولین روز اسپرینت ، 1 آگوست ، تیم برآورد کرده است به میزان Story Point 70 کار برای انجام دادن وجود دارد. در واقع این سرعت برآورد شده اسپرینت می باشد.
- در 16 ام آگوست تیم برآورد کرده است که حدودا از کار 15 Story Point باقی مانده است. خط نقطه چین در وسط نمودار نشان دهنده نرمال ترین وضعیت کار می باشد. تیم اگر برروی این خط حرکت کند خواهد توانست تمام کارهای موجود در اسپرینت را به موقع تکمیل نماید .

ما در محور X ها آخر هفته ها و روز های تعطیل را رد کرده ایم و آن ها را با یک خط صاف مشخص کرده ایم. معمولا در این روزها خط پیشرفت از روز قبل به روز بعد تعطیل حرکت می کند زیرا در این روزها کاری انجام نمی شود. اما اگر در روز های تعطیل هم کار می کنید می توانید آن ها را نیز در محور X ها لحاظ کنید .

علامت خطر تابلوی وظایف

با نگاه مختصر به تابلوی وظایف هر تیم می توان به عملکرد آنها پی برد و این وظیفه مدیر اسکرام می باشد که توجه نماید آیا تیم از وضع خود توسط تابلوی وظایف آگاه هست یا نه؟





اما درباره قابلیت پیگیری؟

اگر به دنبال پیگیری و ره گیری میزان عملکرد تیم هستید من پیشنهاد می کنم هر روزه از تابلوی وظایف عکس بردارید و بدین وسیله می توانید نحوه عملکرد تیم را پیگیری نمایید.

اما اگر قابلیت پیگیری عملکرد برای شما بسیار با اهمیت می باشد ، تابلوی وظایف زیاد برای شما مناسب نیست.

برآورد روزانه در مقابل برآورد ساعتی

در بسیاری از کتاب ها و مقالات اسکرام از برآورد زمانی ساعتی به جای روزانه استفاده می شود. یک راه ساده برای تبدیل واحد زمانی روزانه به ساعتی فرمول ما می باشد : 1 واحد نیرو-روز موثر = 6 واحد نیرو-ساعت موثر.

ما در اوایل از این واحد استفاده می کردیم ولی بدلایلی در بعضی از تیم ها واحد نیرو-ساعت کنار گذاشته شد :

- در برآورد نیرو-ساعت تمایل برای به وجود آوردن وظایف خیلی کوچک مانند 1 ساعت بالا می رود و این نیز باعث به وجود آمدن Micromanagement²⁰ می شود.
- باعث این می شد که افراد فکر کنند که کارهای چند روزه را باید در چند ساعت انجام بدهند. مثلا ضرب دور روز در 6 ساعت می شد 12 ساعت که افراد می کردند که باید همه آنها را در یک روز انجام بدهند.
- استفاده از دو واحد باعث به وجود آمدن سردرگمی می شد : "این برآورد که انجام شده است در چه واحدی است؟".

پس ما برای تمام برآورد های خود از واحد نیرو-روز استفاده می کنیم (که ما آن را بر اساس Story Point بیان می کنیم). کمترین مقدار ما در این واحد 0.5 می باشد و هر مقدار کمتر از 0.5 حذف و با یکی دیگر از داستان ها ترکیب می شود و یا مقدار 0.5 به آن آیتم داده می شود. راه حلی خوب و راحت.

²⁰ نوعی سبک مدیریت می باشد که مدیر کنترل نزدیکی بر همه امور دارد در واقع این نوع مدیریت به مدیریت دستور و کنترل معروف می باشد. مدیریت نخست دستور می دهد که چه کاری باید انجام شود چگونه باید انجام شود و خود بر انجام آن دوباره نظارت می کند. این کلمه معمولا برای لفظ مدیریت نالایق استفاده می شود.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

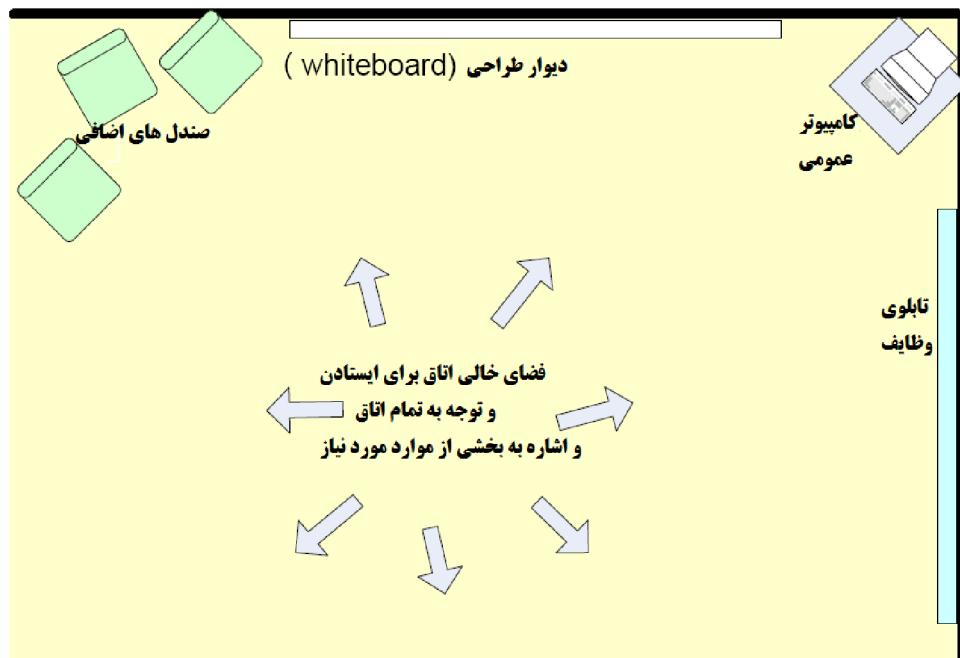
7

اتاق تیم را چگونه مرتب می کنیم

طراحی گوشه ای

من متوجه شده ام که بهترین و با ارزش ترین نوع طراحی برای گفتگوی تیم جا دادن تیم در جلوی تابلوی وظایف می باشد.

برای این منظور، ما سعی می کنیم برای به وجود آوردن فضای خالی در جلوی تابلوی وظایف از طراحی گوشه ای استفاده کنیم.



این نوع طراحی بسیار سودمند می باشد و نوع طراحی دیگری که بتوان به اینگونه به تمام محیط از قبیل دیوار طراحی، کامپیوتر عمومی که در حال ساخت آخرین نسخه برنامه می باشد و تابلوی وظایف و یا هر چیز دیگری در اتاق تیم تسلط داشت وجود ندارد.

"دیوار طراحی" تنها یک وایت برد بزرگ می باشد که می تواند شامل مهمترین مستندات طراحی (نمودار توالی ، پروتوتایپ GUI ، مدل های دامنه و ...) باشد.



عکس بالا نمونه ای از جلسه روزانه اسکرام می باشد که در اتاقی با طراحی گوشه ای ذکر شده انجام شده است.

در عکس بالا نمودار برن داوون بسیار خوب پایین آمده است ، به نظر می رسد عملکرد خوب است . این تیم واقعی است.

نشستن تیم با یکدیگر

افراد از جابجایی بی زار هستند. حداقل در جایی که من کار می کنم اینگونه است. آن ها نمی خواهند تمام وسائل خود را جمع کنند ، کامپیوتر خود را از پریش برق خارج نمایند و به میز جدید بروند و دوباره همه وسائل را در میز جدید قرار بدهند و دوباره کامپیوتر را به برق برنند. "رئیس ، بین اینجا و آنجا فقط 5 متر فاصله است ، به خاطر این باید همه چیز را به هم بزنم ، اجازه بدهید اینجا باشم".

زمانیکه در حال ساخت تیم های موثر اسکرام هستیم هیچ راه دومی جز نشستن تیم با یکدیگر وجود ندارد. اگر فضای کافی برای نشستن تیم با یکدیگر وجود ندارد باید فضایی پیدا کنید حتی اگر این فضا در زیر زمین باشد. فقط سعی کنید که تیم در یکجا و باهم بنشینند.

شاید برای جابجایی و یک جا قرار دادن تیم لازم باشد که آن ها را قول بزنید و شاید آنها غر بزنند ولی به زودی و بعد از یک اسپرینت همه متوجه خواهند شد که یکجا بودن تیم به نفع آن ها بوده است.

حالا , معنی "یکجا" یعنی چه؟ میز ها باید چگونه قرار بگیرند؟ خوب من نظر قطعی در مورد چینش میز ها ندارم ولی موارد زیر می تواند در چینش میز ها و وسائل تیم مفید باشد .

معنی "یکجا نشستن" :

- رسائی صدا^{۲۱}: هر کس باید بتواند بدون اینکه از صدای خود بلند شود و یا داد بزند با هر کدام از افراد تیم صحبت بکند.
- قابلیت دیدن^{۲۲}: هر کس در تیم می تواند افراد دیگر تیم را ببیند. همه افراد تیم می توانند تابلوی وظایف را ببینند. لازم نیست که زیاد به تابلو نزدیک باشند تا همه چیز را بخوانند و فقط دیدن تابلو کفایت می کند.
- تفکیک^{۲۳}: اگر یکدفعه تیم با هم در محوطه خالی اتاق برای یک بحث طراحی به پا خیزند و شروع به بحث کنند این نباید آزار دهنده کسی در خارج از تیم باشد و بالعکس .

تفکیک به این معنی نیست که کلا تیم از سازمان جدا سازی شود. فقط به این معنی که تیم بتواند مکعب خود را داشته باشد و در این مکعب به راحتی با آسایش کار کند و سر و صدای خارج از این مکعب وارد و مزاحم افراد داخل مکعب نشود.

و اگر شما یک تیم مزاحم داشتید باید چکار کنید؟ این اصلا خوب نیست که یک تیم مزاحم بقیه شود. برای جلوگیری از این منظور می توان از تکنولوژی استفاده کرد تا همه سر جای خود بنشینند و از طریق ویدئو کنفرانس با یکدیگر ارتباط برقرار کنند تا مزاحمتی ایجاد نشود.

²¹ Audibility

²² visibility

²³ Isolation

نگه داشتن مالک محصول در بیرون تیم

مالک محصول باید به اندازه کافی به تیم نزدیک باشد تا تیم بتواند سوالات خود را دائماً از او پرسد و مالک محصول هم بتواند بر تابلوی وظایف دید داشته باشد. اما او نباید همراه تیم بنشیند. چرا؟ به این خاطر که او نخواهد توانست خود را از مداخله در امور جزئی و پیاده سازی داستان‌ها حفظ نماید و او مانع ژل شدن تیم خواهد بود.

به طور صادقانه من خودم این مورد را ندیده‌ام که چنین چیزی اتفاق بیفتد. یعنی تجربه این را نداشته‌ام که مالک محصول با خود تیم بنشیند ولی مطلبی که گفته شد بر اساس تجربیات مدیران اسکرام دیگر می‌باشد.

نگه داشتن مدیران و مربیان در بیرون تیم

نوشتن این مطلب برای من بسیار سخت است زیرا من خودم هم مدیر بوده‌ام و هم مربی ...

این وظیفه من بوده است که تا حد امکان با تیم به طور نزدیک کار کنم. من تیم‌ها را برپا کردم، بین تیم‌ها حرکت کرده‌ام، با افراد به صورت جفتی برنامه نویسی کرده‌ام، مدیران اسکرام را مربی‌گری کرده‌ام، جلسه برنامه ریزی اسپرینت را سازماندهی کرده‌ام و غیره. در بازبینی عملکرد بسیاری از افراد اعتقاد داشتند که این خوب بود از اینرو که من تجربه توسعه نرم افزار چابک را داشتم.

اما از طرفی من مدیر ارشد توسعه و نقش مدیر اجرایی را نیز بر عهده داشتم. به این معنی که با ورود این نقش به یک تیم به صورت خودکار از خود-سازماندهی و خود-مدیریت افراد تیم کاسته خواهد شد. "دوستان، رئیس اینجاست، او نظرات و گزینه‌های مختلفی درباره اینکه ما چکار باید بکنیم و چه کسی باید چه کار کند دارد، من از شما می‌خواهم به صحبت‌های آقای رئیس گوش فرا دهید."

نظر من این است: اگر شما یک مربی اسکرام هستید (و یا شاید یک مدیر) تا جایی که امکان دارد همراه تیم شوید ولی فقط یک مدت محدود و نه برای همیشه و بعد این مدت خارج شوید و به تیم اجازه بدهید تا کامل ژل و خود-سازمانده شوند. می‌توانید برای کنترل تیم در دموی اسپرینت به آنها ملحق شوید و یا در جلسات روزانه اسکرام به سخنان آنها گوش دهید. اگر شما مسئله‌ای می‌بینید که

می تواند باعث ارتقاء تیم شود مدیر اسکرام تیم را فرا بخوانید و او را مربی گری نمایید و البته نه در جلوی تیم. ایده دیگر این هست که می توانید در جلسات بازمینی اسکرام شرکت کنید.

برای ایجاد تیم های خوب بازده , مطمئن شوید که آن ها تمام چیز هایی را که نیاز داشتند را گرفته اند , سپس کاملاً آن ها را رها کنید (به جز ارائه دموی اسپرینت).

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

8

چگونه جلسه روزانه اسکرام را برگزار می کنیم

جلسات روزانه اسکرام ما نسبت به بقیه جلسات اسکرام بیشتر بر اساس دستورالعمل انجام می شود. در یک مکان و زمان مقرر شروع می شود. جلسه روزانه اسکرام را در اتاق تیم و درست جلوی تابلوی وظایف برگزار می کنیم.

ما این جلسه را به صورت ایستاده انجام می دهیم تا ریسک گذشتن از 15 دقیقه زمان جلسه کاهش یابد.

چگونه تابلوی وظایف را به روزرسانی می کنیم

به طور معمول تابلوی وظایف را طی جلسه اسکرام روزانه به روزرسانی می کنیم. به صورتی که هر کس توضیح می دهد که دیروز چه کاری انجام داده است و امروز چه کاری می خواهد انجام دهد و وظایف را به همان منوال توضیح داده به جلو حرکت می دهد. زمانیکه او موارد طرح ریزی نشده را توضیح می دهد او آیتم را در قسمت موارد طرح ریزی نشده تابلوی وظایف قرار می دهد. او می تواند در صورت نیاز برآوردهای زمان قبلی را خط بزند و برآورد های جدید را یادداشت نماید.

Write
failing
2d test

Write
failing
2d test
3d

Write
failing
2d test
3d 1d

بعضی از تیم ها بدین صورت عمل می کنند که قبل از جلسه هر کس تابلوی وظایف را نسبت به کارهای خود آپدیت می نماید و در زمان جلسه فقط توضیح می دهند. شما هم می توانید از هر کدام از روش های موجود استفاده کنید.

بدون توجه به اینکه از چه فرمتی برای بک لاگ اسپرینت استفاده می کنید، باید توجه داشته باشید که همه افراد در آپدیت بک لاگ اسپرینت تعامل داشته باشند. در بعضی تیم ها فقط مدیر اسکرام نقش نگه دارنده بک لاگ اسپرینت را بازی می کند و او از تیم سوال می پرسد که چه کارهایی را انجام داده اید و چه کاری را می خواهید انجام بدهید و او بر این اساس بک لاگ اسپرینت را آپدیت می کند. معایب این روش می تواند موارد ذیل باشد:

- مدیر اسکرام به جای اینکه زمان بیشتری برای پشتیبانی از تیم و رفع موانع صرف نماید باید وقت خود را صرف مدیریت آیتم های بک لاگ نماید.
 - اعضای تیم از وضعیت اسپرینت بی خبر خواهند شد و اسپرینت بک لاگ چیزی نخواهد بود که آنها نگران آن باشند. این بی خبری باعث کاهش سطح چابکی و کاهش توجه تیم خواهد شد.
- اگر بک لاگ اسپرینت خوب طراحی شده باشد باید همه اعضای تیم بتوانند به راحتی آن را آپدیت نمایند.

بعد از جلسه روزانه اسکرام شخصی بلافاصله Story Point های باقی مانده را محاسبه و در نمودار Burndown اعمال می کند.

مواجهه با دیر کنندگان

بعضی تیم های اسکرام برای خود یک جعبه سکه دارند. زمانی که کسی برای جلسه دیر می کند حتی 1 دقیقه باید مبلغ ثابتی را به درون جعبه بیندازد. و هیچ سوالی در این مورد پرسیده نمی شود. حتی اگر قبل از جلسه تماس بگیرید و اعلام کنید که امروز دیر خواهید کرد باز هم باید مبلغ را بپردازید.

فقط زمانی می توانید از پرداخت بگریزید که بهانه خوبی مانند وقت دکتر و یا عروسی و یا چنین چیزی داشته باشید.

پول جمع شده صرف رویدادهای عمومی خواهد شد. مثلاً صرف خرید همبرگر برای شب هایی می شود که تیم دور هم جمع است.

این روش خوب است ولی در بعضی تیم ها نیاز نیست که انجام شود. از این روش زمانی استفاده می شود که معمولاً افراد دیر می کنند و لازم نیست که همیشه از این روش استفاده کنید.

مواجهه با "من نمی دانم که چه چیزی را امروز انجام بدهم"

این مورد غیر عادی نیست که در جلسه روزانه اسکرام بشنوید "من دیروز فلان کار را انجام دادم ولی امروز نمی دانم باید کدام کار را شروع کنم؟!".

اگر من در آن تیم، مدیر اسکرام باشم، ما از آن نفر رد می شویم و اجازه می دهیم که نفر بعدی صحبت کند ولی آن نفراتی که کار ندارند را به یاد می سپاریم. بعد از اینکه همه صحبت های خود را تمام کردند، همراه تیم تمام موارد را از بالا به پایین بررسی می کنیم تا ببینیم که آیا همه هماهنگ هستند و آیا همه معنی داستان ها را می دانند. از افراد دعوت می کنم که اگر نیاز است وظایف بیشتری برای داستان ها استخراج شود و بر روی داستان مربوطه نصب شود. بعد از این به سراغ کسانی می روم که نمی دانستند چه کاری باید انجام بدهند و از آن ها سوال می پرسم "بعد از اینکه ما تمام داستان ها و وظایف تابلو را بررسی کردیم، آیا ایده ای به نظرتان رسید که بخواهید امروز انجام بدهید؟". به طور امیدوارانه باید توانسته باشند.

اگر هنوز نه، امکان انجام برنامه نویسی جفتی را بررسی می کنم. اگر امکان این وجود داشته باشد برای کمک به یکی از نیروی های کار، این برنامه نویس را جفت او تعیین خواهم کرد. اگر این امکان هم نبود به راه حل بعدی دقت نمایید.

مدیر اسکرام: "چه کسی می خواهد دموی اسپرینت را به ما نشان دهد؟"

تیم: سکوت ابهامانه.

مدیر اسکرام: "مگر همه کارها را تکمیل نکرده ایم؟"

تیم: "اوووم ... نه".

مدیر اسکرام: "چرا؟ من فکر می کردم که همه چیز تمام شده! چه چیزهایی باقی مانده؟".

تیم: "خوب ما هنوز سرور آزمایش برای اجرای برنامه نداریم و اسکریپت ساخت نیز از بین رفته است".

مدیر اسکرام: "آها" (این دو وظیفه را به تابلوی وظایف اضافه کنید). "جو و لیزا (آن هایی که نمی

دانستند چه کار باید بکنند)، امروز شما چگونه می توانید به ما کمک کنید؟"

جو: "خوب ... حدس می زنم باید سعی کنم از جایی یک سرور آزمایش پیدا کنم".

لیزا: "... من هم سعی خواهم کرد که اسکریپت ساخت را درست نمایم".

اگر این وضعیت در اواخر اسپرینت اتفاق بیفتد، با اضافه کردن این دو وظیفه خواهید توانست که به هدف اسپرینت دست بیابید و دمو اسپرینت را آماده کنید. اما اگر این مشکل در وسط اسپرینت اتفاق بیفتد؟ به تیم تبریک بگویید که کار خود را عالی انجام داده اند و همه چیز را قبل از موعد مقرر تمام کرده اند. از داستان های قسمت "داستان های بعدی" تابلوی وظایف می توانید چند آیتم را وارد بک لاگ اسپرینت نمایید. ولی به یاد داشته باشید که باید این مسئله را به اطلاع مالک محصول برسانید.

اما اگر تیم به هدف اسپرینت دست نیافته باشد و هنوز جو و لیزا ندانند که چه کار مفیدی می توانند انجام بدهند، چطور؟ من در این حالت از روش های زیر استفاده می کنم:

- **شرمنده کردن:** "اگر شما هیچ ایده ای ندارید که چگونه امروز به تیم کمک کنید، پیشنهاد می کنم به خانه بروید و یا کتاب بخوانید و یا همچنین چیزی. یا در جایی ساکت بنشینید تا کسی شما را صدا کند."
- **مدرسه - قدیمی:** تعیین کردن یک وظیفه ساده برای آنها.
- **فشار جفتی:** "از این لحظه وقت هر دوی شما آزاد می باشد تا زمانی که با مشورت با یکدیگر به یک راه حلی برسید که چگونه می توانید تیم را در دست یابی به هدف کمک نمایید."
- **بردگی:** "خوب شما می توانید تیم را برای رسیدن به هدف کمک کنید مثلاً با آوردن قهوه، ماساژ دادن آنها و یا هر چیز دیگری که در طی روز ما بخواهیم". خیلی زود متوجه خواهید شد که جو و لیزا چقدر سریع برای خود یک داستان فنی به جای ماساژ دادن پیدا خواهند کرد.

اگر شخصی وجود داشت که دائماً شما را در چنین وضعیتی قرار می داد و معمولاً نمی داست که چه کار باید انجام دهد. بهتر است او را از تیم خارج و مورد مریگیری جدی قرار دهید. اگر پس از این دوباره مشکل پابرجا ماند باید اهمیت وجود این شخص برای شما و تیم را مورد ارزیابی قرار دهید.

اگر وجود او برای تیم خیلی ضروری نبود، سعی کنید او را از تیم خود حذف کنید. اگر وجود او بسیار ضروری بود، سعی کنید او را با کس دیگری جفت نمایید کسی که بتواند چوپان او باشد. شاید جو یک برنامه نویس و معمار فوق العاده است و او ترجیح می دهد که همیشه کسی به او بگوید که چه کاری انجام بدهد. خوب است. نیکلاس را به عنوان چوپان دائمی جو تعیین کنید و یا وظیفه چوپانی جو را خودتان قبول کنید. اگر جو واقعا برای تیم مهم باشد فعالیت های او برای تیم با ارزش خواهد بود.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

9

چگونه دموی اسپرینت را انجام می دهیم

دموی اسپرینت (یا مرور اسپرینت) یکی از بخش های مهم اسکرام می باشد که بعضی از افراد آن را کم ارزش در نظر می گیرند.

"آیا واقعا ما مجبوریم که جلسه ارائه دمو داشته باشیم؟ خیلی جالب به نظر نمی رسد"

"ما زمان برای آماده کردن یک دموی لعنتی را نداریم"

"من وقت این را ندارم که به دیدن دموی تیم های دیگر بروم."

چرا ما اصرار می کنیم که همه اسپرینت ها با ارائه یک دمو تمام شوند؟

ارائه یک دمو برای اسپرینت می تواند تاثیرات زیادی داشته باشد.

- تیم برای کارهای انجام داده صاحب اعتبار خواهد شد. آنها احساس خوبی خواهند داشت.
- افراد دیگر یاد خواهند گرفت که تیم شما چه کاری انجام می دهد.
- دمو باعث جذب بازخوردهای اساسی از طرف ذینفعان خواهد شد.
- ارائه دمو یک رویداد اجتماعی می باشد. در این جلسه تیم های دیگر هم حضور خواهند داشت و در مورد دمو بحث خواهند کرد که می تواند با ارزش باشد.
- ارائه دمو باعث خواهد شد که افراد واقعا کارها را تکمیل نمایند. بدون دمو معمولا کارها تا مرز 99% تکمیل می شوند. اما با ارائه دمو شاید تعداد داستان های تکمیل شده کمتر شوند ولی همه آنها واقعا تکمیل شده اند که قابل ارائه می باشند.

اگر حتی یک تیم کارها را به صورت جدی تکمیل ننماید و جلسه ارائه دمو را جدی نگیرد. حتما در زمان ارائه دمو شاهد بروز چندین خطا و باگ خواهند بود. این باعث خواهد شد که مدعوین جلسه از

اینکه به جلسه چنین دمویی آمده اند پشیمان شوند و این باعث شرمندگی ارائه کنندگان دمو خواهد بود.

این می تواند باعث لطمه دیدن تیم شود ولی این لطمه همانند داروهای بد مزه پزشکان خواهد بود. مطمئن باشید تیم در ارائه دموی بعدی تلاش خود را به 100٪ خواهند رساند تا یک ارائه فوق العاده داشته باشد. شاید اعضای تیم با خود بگویند "به جای اینکه 5 مورد را ناقص درست کنیم بهتر است 2 داستان را به صورت 100٪ و کامل تکمیل نماییم". و این از تاثیرات خوب ارائه دمو می باشد.

چک لیست برای دموی اسپرینت

- مطمئن شوید که به طور واضح هدف اسپرینت را ارائه می دهید. اگر در جلسه افرادی هستند که هیچ چیز درباره محصول شما نمی دانند بهتر است کمی به طور خلاصه در مورد محصول توضیح دهید.
- زمان زیادی را برای آماده سازی دمو صرف نکنید بخصوص برای ساخت ارائه های پر زرق و برق. اضافات را قیچی کنید و فقط بر روی ارائه کدهای کارآی واقعی متمرکز شوید.
- سرعت ارائه را بالا نگه دارید – در زمان آماده سازی دمو در فکر این باشید که سرعت ارائه بالا باشد و نه اینکه زیبا باشد.
- ارائه دمو را به جای فنی گرایی و اصلاحات فنی، تجارت گرا آماده کنید. هدف این است که بگوییم "ما چه کاری انجام داده ایم" به جای اینکه "چطور انجام داده ایم".
- اگر امکان دارد، اجازه بدهید شرکت کنندگان خودشان از محصول استفاده کنند.
- در این جلسه فقط چند باگ رفع شده و یا چند داستان کوچک را نشان ندهید. می توانید آنها را به صورت کلامی ذکر کنید ولی هدف اصلی جلسه نمایش داستان ها و امکانات مهم می باشد.

مواجهه با مسائل غیر قابل ارائه در نمایش دمو

نفری از تیم: "من قصد ندارم این آیتم را برای ارائه دمو آماده کنم. زیرا این آیتم قابل دمو شدن نیست. داستان این می باشد که "بهبود قابلیت گسترش سیستم، تا سیستم بتواند تا 10,000 کاربر همزمان را اداره کند". من نمی توانم 10,000 کاربر را برای شبیه سازی کاربران سیستم دعوت کنم!؟"

مدیر اسکرام: "آیا این کار را به صورت کامل تکمیل کرده ای؟"

نفری از تیم: "بلی - کاملاً".

مدیر اسکرام: "از کجا می دانی؟"

نفری از تیم: "من سیستم را در محیط تست پرفرمنس راه اندازی کردم, با 8 لود سرور شروع شد و حالت پر کابر برای سیستم شبیه سازی شد."

مدیر اسکرام: "آیا مدرک و یا نشانه ای که سیستم خواهد توانست از 10,000 کاربر به صورت همزمان پشتیبانی کند, داری؟"

نفری از تیم: "بلی. در تست ثابت شد که سیستم تا 50,000 کاربر پشتیبانی می کند."

مدیر اسکرام: "از کجا می دانی؟"

نفری از تیم: "خوب, من این را گزارش دادم. شما خودتان می توانید برگه گزارش تست را مشاهده کنید و ببینید که چگونه سیستم تست شده است و چند درخواست همزمان ارسال شده است." مدیر اسکرام: "بسیار عالی. پس این آیتم دمو دارد. فقط کافی است در زمان ارائه دمو این گزارش را به مدعوین نشان دهیم. این بهتر از هیچ است!"

نفری از تیم: "آه ... آیا این کافی می باشد؟ گزارش من خیلی زشت بود و نیاز به آراسته کردن دارد."

مدیر اسکرام: "بسیار خوب. ولی وقت زیادی بر روی آن صرف نکن. نیاز نیست خیلی زیبا شود فقط آگاهی دهنده باشد کافی است."

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

10

چگونه بازیابی اسپرینت را انجام می دهیم

چرا اصرار داریم که همه تیم ها بازیابی اسپرینت را انجام دهند؟

مهمترین مسئله در مورد بازیابی این است که جلسه بازیابی اتفاق بیفتد.

بدلایلی تیم ها بعضا از انجام جلسه بازیابی سر باز می زنند. بدون هیچ رودر بایستی اکثر تیم های ما بدون توجه به ارزش این جلسه آن را رد می کنند و اسپرینت بعدی را شروع می کنند. البته شاید این یک مسئله فرهنگی در سوئد باشد، مطمئن نیستم.

با این حال به نظر می رسد که همه با فواید بی نهایت بازیابی موافق هستند. در واقع من اعتقاد دارم که بازیابی در اسکرام دومین کار پر اهمیت (بعد از جلسه برنامه ریزی اسپرینت) می باشد زیرا شما بهترین فرصت را برای بهبود می یابید.

البته نیاز نیست که جلسه بازیابی را برای دست یابی به ایده های خوب برگزار کنید. می توانید این کار را در وان حمام تان نیز انجام دهید! اما آیا تیم ایده شما را قبول خواهد کرد؟ شاید، اما بر طبق تجربه، ایده هایی بیشتر خریدار خواهند داشت که آن ها از دل تیم و با شرکت همه اعضای تیم پدید آمده باشند. جلسه بازیابی اسکرام می تواند پدیدار کننده این ایده های تیمی باشد. بدون بازیابی شما بعدا متوجه خواهید شد که تیم همان اشتباهات سابق را دوباره و دوباره انجام می دهد و این روند نیز ادامه دار می باشد.

جلسات بازیابی را چگونه سازماندهی می کنیم

معمولا قالب کلی این جلسات کمی متغیر می باشد ولی ما در کل به مانند زیر عمل می کنیم:

- 1-3 ساعت برای جلسه (بسته به پیش بینی مقدار بحث) اختصاص می دهیم.
 - شرکت کنندگان: مالک محصول، تمام تیم و من خودم.
 - ما به یک اتاق بسته می رویم مانند یک جای دنج یا پاسیو و یا جایی مثل آن. جایی که بتوان یک بحث طولانی بدون مزاحمت داشت.
 - معمولاً جلسه بازبینی را در اتاق تیم انجام نمی دهیم. بدلیل اینکه افراد تمایل به خیره شدن به دور و اطراف اتاق خود را دارند.
 - فردی به عنوان منشی منصوب می شود.
 - مدیر اسکرام بک لاگ اسپرینت را نشان می دهد و به کمک تیم اسپرینت را جمع بندی می نماید.
 - ما از اصطلاح "راند" استفاده می کنیم. هر کس یک راند فرصت پیدا می کند تا بدون ایجاد وقفه صحبت نماید: او فکر می کند چه چیزهایی خوب بودند، چه چیزهایی می توانند بهتر انجام شوند و چه چیزهای را می خواهد که کلا در اسپرینت بعدی متفاوت انجام شوند.
 - به سرعت های برآورد شده و سرعت واقعی نگاه می کنیم. اگر اختلاف فاحش باشد سعی می کنیم مشکل را مورد آنالیز قرار دهیم.
 - در اواخر جلسه مدیر اسکرام سعی می کند که بحث های انجام شده را در قالب "چه چیزهای بهتری می توان در اسپرینت بعدی انجام داد" جمع بندی کند.
- جلسه بازبینی ما معمولاً زیاد دارای ساختار نیست. اما هدف و خروجی اصلی در هر ساختاری این خواهد بود که "چه چیزهای بهتری می توان در اسپرینت بعدی انجام داد".

عکس پایین نشان دهنده جلسه بازبینی اخیر ما می باشد:



- **خوب (Good) :** اگر همان اسپرینت را دوباره انجام دهیم باز هم این کارها را انجام خواهیم داد.
- **می تواند بهتر انجام شود :** اگر همان اسپرینت را دوباره انجام بدهیم این کارها را متفاوت انجام خواهیم داد.
- **بهبود :** ایده های واقعی در مورد اینکه چگونه می توانیم خودمان را در آینده بهبود دهیم.

ستون ها 1 و 2 مربوط به گذشته هستند ولی ستون 3 مربوط به آینده می باشد.

بعد از هم فکریه تیم در مورد تمامی این آیتم ها ، برای اینکه معلوم شود کدام یک از این بهبود ها در اسپرینت بعدی باید اعمال شوند از رای گیری نقطه ای استفاده می شود. هر یک از اعضا 3 آهن ربای کوچک دریافت می کنند.

از اعضای تیم دعوت می شود به مواردی که با اهمیت می دانند و می خواهند در اسپرینت بعدی آن ها بهبود یابند ، به آنها رای دهند. آنها می توانند هر 3 آهن ربا را بر روی یک آیتم بچسبانند و یا هر کدام را بر روی یک آیتم.

آنها 5 آیتم بر اساس رای های موجود را انتخاب می کنند تا در اسپرینت بعدی برای ایجاد بهبود اعمال نمایند.

نکته اساسی اینجا است که نباید در هر اسپرینت بهبودهای زیادی با هم اعمال شوند. در هر اسپرینت سعی کنید چند بهبود را وارد نمایید تا موثرتر باشند .

گسترش درس ها بین تیم ها

اطلاعات بدست آمده از جلسات بازیابی معمولاً بسیار با ارزش می باشند. جلسه بازیابی در مورد این نیست که فقط یک تیم چگونه می تواند کارها را در اسپرینت بعدی بهتر انجام دهد ، این مفهوم فراتر از حوزه کاری یک تیم می باشد.

روش ما برای اجرای این هدف بسیار ساده می باشد. یک نفر (که معمولاً من هستم) در همه جلسات بازیابی اسکرام (حتی تیم های دیگر) شرکت می کند و در واقع او نقش پل اطلاعاتی را برای تیم ها بازی می کند.

یک راه دیگر این می باشد که هر تیم گزارش مربوطه به بازیابی اسکرام را منتشر نماید. ما انجام چنین چیزی را تجربه داریم ولی متأسفانه شاهد این بودیم که افراد زیاد مایل به خواندن این گزارشات نیستند.

اصول مهم برای شخص "پل ارتباطی":

- او باید شنونده خوبی باشد.
- اگر جلسه بازیابی خیلی ساکت بود، او باید با پرسیدن سوالات ساده اما کمک کننده، به باز شدن بحث گروه کمک نماید. برای مثال "اگر شما قرار بود این اسپرینت را از اول انجام دهید، چه چیزهایی را متفاوت انجام خواهید داد؟".
- او باید مشتاق گذراندن وقت خود در جلسات بازیابی تمام تیم ها باشد.
- او باید دارای یک انرژی مثبت باشد تا بتواند پیشنهادات بهبود دهنده خود را از خارج از تیم به داخل تیم ارائه دهد.

این کار به روش های بهتری هم قابل انجام هست، در این مورد اگر راه بهتری پیدا کردید من را نیز خبر کنید.

ایجاد تغییر یا ایجاد نکردن تغییر

به این نتیجه رسیده ایم که در اکثر اوقات فقط شناسایی مشکل تنها برای حل آن به طور خودکار در اسپرینت کافی می باشد. بخصوص اگر شما نتیجه بازیابی را در اتاق تیم نصب کرده باشید (که معمولاً یادمان می رود - شرم بر ما). هر تغییری که انجام می گیرد دارای هزینه ای می باشد، برای جلوگیری از هزینه روش انجام ندادن هیچ کاری را بررسی کنید و امیدوار باشید که در اسپرینت بعدی به صورت خودکار مشکل حل خواهد شد. (یا حداقل کوچکتر خواهد شد).

مثال ("ما بسیار کم در داخل تیم ارتباط داریم...") از مشکل هایی می باشد که بهترین راه حل انجام ندادن هیچ کاری می باشد. یعنی با شناسایی بعضی از مشکلات می توان به حل شدن خودکار آن مشکلات بدون هیچ هزینه ای دست یافت.

اگر قرار باشد هر وقت که کسی از چیزی شکایت کرد یک تغییر جدید به وجود آوریم ، این بسیار با هزینه خواهد بود.

اتفاق هایی که ممکن است در بازبینی اتفاق بیفتد

در زیر مثال هایی از اتفاق هایی که ممکن است در جلسه بازبینی اتفاق بیفتند موجود می باشند:

"باید زمان زیادی را برای شکستن داستان ها به داستان ها کوچک تر و همینطور به وظایف صرف کنیم"

این مشکل معمولا اتفاق می افتد. هر روزه در جلسه روزانه اسکرام ، اعضای تیم اذعان می کنند "من واقعا نمی دانم که امروز چه کاری باید انجام دهم!؟". بنابراین بعد از هر جلسه روزانه اسکرام زمانی برای پیدا کردن وظیفه جهت انجام دادن صرف می شود.

اقدام نوعی : هیچ چیز. تیم فقط می تواند به صورت خودکار این مشکل را حل کند. برای حل این قضیه می توان کمی به زمان جلسه برنامه ریزی اسپرینت اضافه کرد تا بتوان وظایف زیادی را استخراج کرد.

"اختلالات و ناراحتی های بیرونی خیلی زیاد"

اقدام نوعی :

- از تیم بخواهید در اسپرینت بعدی ضریب توجه خود را پایین تر در نظر بگیرد تا بتواند برنامه ریزی واقعی تری نسبت به شرایط اتخاذ نمایند.

- از تیم بخواهید که اختلالات موجود را به یاد بسپارند: چه کسی اختلال به وجود می آورد؟ مدت آن چه قدر است؟ کی ها معمولا به وجود می آید؟ معمولا بعدا می توان آن را راحت تر رفع کرد.
- از تیم بخواهید تمام این اختلالات را به مدیر اسکرام و یا مالک محصول اطلاع دهند.
- از تیم بخواهید شخصی را به عنوان گلر منتصب نماید، رفع تمام اختلالات بر عهده این شخص خواهد بود و تیم می تواند به راحتی بر روی هدف خود تمرکز نماید.

"ما تعهد خود را نتوانستیم به جا آوریم و فقط نصف مواردی که قول داده شده بود انجام شد."

اقدام نوعی: هیچ چیز. تیم در اسپرینت بعدی تعهد خود را به جا خواهد آورد و یا حداقل به این فضاقت نخواهد بود که نصف کارها فقط انجام شده باشد.

"محیط کاری ما بسیار پر سروصدا و شلوغ می باشد"

اقدام نوعی:

- سعی کنید یک محیط بهتر ایجاد کنید یا تیم را به جای دیگر انتقال دهید. (برای مثال یک اتاق در هتل برای تیم اجاره کنید).
- اگر ممکن نیست، از تیم بخواهید در اسپرینت های بعدی ضریب توجه خود را به خاطر محیط پایین در نظر بگیرند. این معمولا کار ساز خواهد بود و مدیریت برای رفع این موضوع مجبور به ایجاد یک محیط بهتر خواهد شد.

خوشبختانه من هیچ وقت لازم نشده است که تیم را از جایی منتقل کنم ولی اگر لازم باشد به خاطر تیم این کار را انجام خواهم داد.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

11

زمان انقطاع بین اسپرینت ها

در زندگی واقعی نمی توانید دائما اسپرینت ها را پشت سر هم انجام بدهید. باید مابین اسپرینت ها استراحتی انجام بدهید. اگر پشت سرهم انجام دهید باعث تحلیل رفتن نیروی کار خواهید شد.

اسپرینت ها معمولا فشار زیادی ایجاد می کنند. به عنوان توسعه گر هیچ زمانی بین اسپرینت ها نخواهید داشت و مجبور خواهید بود هر روز ایستاده به دیگران بگویید که دیروز چه کاری انجام دادید و امروز چه کاری می خواهید انجام بدهید. در این حالت بعضی ها مستعد این خواهند شد که بگویند "من بیشتر وقت خود را صرف گشت و گذار میان سایت و وبلاگ های اینترنتی کردم".

دلیل دیگر، علاوه بر استراحت این خواهد بود که بعد از ارائه دمو و جلسه بازبینی، هم تیم و هم مالک محصول پر از اطلاعات خواهند بود. اما برای سودمندی این اطلاعات و تبدیل آن به ایده های خوب نیاز به کندوکاو خواهند داشت. اگر قرار باشد بعد از اتمام اسپرینت، اسپرینت بعدی شروع شود زمانی برای اینکار به وجود نخواهد آمد.

برنامه ریزی نامناسب :

دوشنبه

10-09 : دمو اسپرینت 1

11-10 : بازبینی اسپرینت 1

16-13 : برنامه ریزی اسپرینت 2

ما سعی داریم روش هایی برای ایجاد زمان بین اسپرینت ها معرفی کنیم بخصوص بین جلسه بازبینی و جلسه برنامه ریزی اسپرینت بعدی. به هر حال ما همیشه در این مورد موفق نبوده ایم.

حداقل ما سعی می کنیم که مطمئن شویم جلسه بازبینی و جلسه برنامه ریزی اسپرینت بعدی در یک روز اتفاق نمی افتند. هر کس باید حداقل یک شب بدون اسپرینت را داشته باشد تا بتواند جلسه برنامه ریزی خوبی را سپری کند.

برنامه ریزی بهتر :

سه شنبه 13-9 : برنامه ریزی اسپرینت 2	دوشنبه 10-09 : دموی اسپرینت 1 11-10 : بازبینی اسپرینت 1
--	--

بسیار بهتر :

دوشنبه 13-9 : برنامه ریزی اسپرینت 2	یکشنبه	شنبه	جمعه 10-09 : دموی اسپرینت 1 11-10 : بازبینی اسپرینت 1
---	---------------	-------------	--

یک روش برای انجام این ، تعیین "روزهای آزمایش" می باشد. (یا هر چیزی که شما آن را می نامید). در این روز به توسعه گران اختیار داده می شود که هر کاری که می خواهند انجام بدهند (من این را از گوگل الهام گرفته ام). برای مثال آشنایی با یک ابزار خاص ، پیدا کردن API ، مطالعه برای اخذ گواهی نامه ، بحث در موردی با همکاران ، یا کدنویسی در پروژه های سرگرمی و غیره.

هدف ما این است که این روزهای آزمایش در بین دو اسپرینت قرار بگیرند. با قرار دادن این روز مابین دو اسپرینت باعث به وجود آمدن وقفه بین اسپرینت ها خواهیم شد که این علاوه بر ایجاد زمان برای به روز شدن توسعه گران باعث شادابی توسعه گران نیز خواهد شد.

بهترین ؟

دوشنبه 13-9 : برنامه ریزی اسپرینت 2	یکشنبه	شنبه	جمعه روز آزمایش	پنج شنبه 10-09 : دموی اسپرینت 1 11-10 : بازبینی اسپرینت 1
--	---------------	-------------	------------------------------	--

هم اکنون ما در هر ماه یک روز آزمایش داریم. اولین جمعه هر ماه را برای این روز مشخص کرده ایم. چرا پس این بین اسپرینت ها نیست؟ خوب، من احساس می کردم که عده ای این روز را جدی نمی گرفتند و فکر کردم که اگر تمام شرکت با هم در روز مشخص این روز را صرف مطالعات و آزمایش قرار بدهند موثرتر خواهد بود.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

12

چگونه برنامه ریزی Release و قراردادهای قیمت بسته را انجام

می دهیم

بعضی مواقع نیاز است که برنامه ریزی فراتر از یک اسپرینت انجام شود. به طور نوعی در بر خورد با قراردادهای قیمت بسته که مجبوریم یا برنامه ریزی بسیار جلوتر را انجام بدهیم و یا باید یک قرار داد پراز ریسک را امضا کنیم که حق تاخیر نداریم.

به طور نوعی ، برنامه ریزی Release می خواهد جواب سوال "کی" را بدهد. یعنی کی ما قادر خواهیم بود اولین نسخه از سیستم را ارائه بدهیم.

اگر واقعا می خواهید برنامه ریزی Release را یادگیرید این فصل را رد کنید و کتاب "برآورد و برنامه ریزی"²⁴ مایک کان²⁵ را بخوانید. نسخه برنامه ریزی Release من بسیار ساده تر از کتاب عنوان شده می باشد.

سرحد پذیرش را تعیین کنید

معمولا در قراردادهای بسته علاوه بر بک لاگ معمول ، مالک محصول یک لیست با عنوان سرحد پذیرش را نیز آماده می کند. این لیست به این معنی می باشد که اجزای این لیست حداقل های مورد نظر مالک محصول برای پذیرش انجام قرار داد می باشد. یعنی در صورت انجام تمام این موارد سرحد ، قرار داد انجام شده است.

نمونه ای از قوانین سرحد پذیرش :

²⁴ Agile Estimating And Planning

²⁵ Mike Cohn

- تمام موارد با ارزش اهمیت ≤ 100 بایستی در نسخه یک انجام شوند در غیر اینصورت قابل پذیرش نیست.
- تمام موارد با اهمیت 50 – 99 باید شامل نسخه یک باشند. اما ما این حق را خواهیم داشت که بعدا آن ها را از جریان نسخه یک خارج نماییم.
- آیتم های با اهمیت 25-49 ضروری هستند اما می توان آن ها را در طی ارائه 1.1 انجام داد.
- آیتم های با اهمیت کمتر از 25 خیلی ضروری نیستند و می شود به بعد موکول شوند .

در زیر عکسی از بک لاگ محصول می باشد که با قوانین ذکر شده رنگی شده است :

اهمیت	نام
130	banana
120	apple
115	orange
110	guava
100	pear
95	raisin
80	peanut
70	donut
60	onion
40	grapefruit
35	papaya
10	blueberry
10	peach

5 تای اول (نارنجی): آنهایی که می بایستی در نسخه یک باشند.

4 تای دوم (زرد): آنهایی که باید در نسخه یک باشند.

4 تای آخر (سبز): آنهایی که بعدا می توانند انجام شوند.

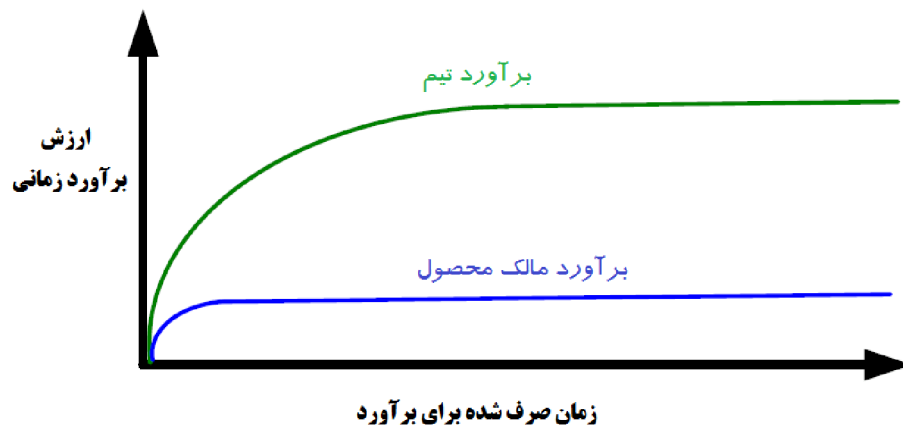
اگر ما بتوانیم در زمان قرار داد موارد نارنجی و زرد را که همان ضرب العجل پروژه محسوب می شوند ارائه بدهیم ایمن خواهیم ماند و اگر زمان این را داشته باشیم که موارد سبز را نیز انجام بدهیم آنها به عنوان جایزه در نظر گرفته می شوند.

برآورد زمانی آیتم های بسیار مهم

برای انجام برنامه ریزی Release, مالک محصول حداقل باید داستان های موجود در قرار داد را همراه با تیم برآورد نمایند. درست مانند برنامه ریزی اسپرینت. یک تلاش گروهی برای برآورد داستان ها مابین تیم و مالک محصول. مالک محصول داستان ها و دامنه آنها را شرح می دهد و تیم بر اساس درک خود داستان ها را برآورد می کند.

یک برآورد زمانی با ارزش خواهد بود که نزدیک به واقعیت باشد و زمانی کم ارزش خواهد شد که دور از واقعیت باشد.

در شکل زیر ارزش برآورد زمانی بر اساس اینکه چه کسی برآورد را در چه مدت زمانی انجام داده است نشان داده می شود:



تمام شواهد گویای این مطلب می باشد که :

- اجازه بدهید تا تیم برآورد را انجام بدهد.
- زمان زیادی را برای برآورد صرف نکنید.
- مطمئن شوید که آنها برآورد را یک برآورد خام در نظر بگیرند و نه یک تعهد.

معمولا مالک محصول اعضای تیم را در یک اتاق جمع می کند و به آنها می گوید که هدف ما برآورد 20 آیتم بالای بک لاگ محصول می باشد. او از میان داستان ها حرکت می کند و به تیم اجازه می دهد که هر کدام را برآورد نمایند. مالک محصول همانند برنامه ریزی اسپرینت جوابگویی سوالات تیم خواهد بود تا جلوی کج فهمی ها گرفته شود. معمولا برای هر آیتم چگونگی دمو (جهت تعیین دامنه) معین می شود.

این جلسه همانند تمام جلسات اسکرام باید در یک زمان معین تمام شود. زیرا تیم علاقه دارد که زمان زیادی بر روی برآورد داستان ها صرف کند.

اگر مالک محصول خواهان این باشد که زمان بیشتری بر روی برآورد کردن صرف شود باید یک جلسه دیگر زمانبندی و همه را دوباره دعوت نماید.

نمونه ای از خروجی این جلسه

اهمیت	نام	برآورد
130	banana	12
120	apple	9
115	orange	20
110	guava	8
100	pear	20
95	raisin	12
80	peanut	10
70	donut	8
60	onion	10
40	grapefruit	14
35	papaya	4
10	blueberry	
10	peach	

برآورد سرعت

بسیار خوب , ما تا کنون یک سری برآورد خام بر روی داستان های مهم انجام داده ایم. گام بعدی تعیین سرعت متوسط به ازای هر اسپرینت می باشد.

به این معنی که ما نیاز داریم در مورد ضریب توجه خودمان تصمیم بگیریم.

ضریب توجه به طور ساده "چقدر از وقت تیم صرف توجه به داستان های متعهد شده اسپرینت می شود؟". این زمان هیچ وقت 100٪ نیست از اینرو که تیم وقت خود را برای موارد طرح ریزی نشده، سوئیچ موارد، کمک به سایر تیم ها، چک کردن ایمیل ها، تعمیر کامپیوتر های خراب شده، بحث سیاست در آشپزخانه و غیره صرف می کند.

فرض کنید که ضریب توجه را 50٪ تعیین کردیم (این بسیار کم است و معمولا ما 70٪ می گیریم) و طول اسپرینت های ما 3 هفته خواهد بود (15 روز) و اندازه تیم ما 6 نفر می باشد.

در هر اسپرینت 90 نیرو - روز موجود می باشد ولی بدلیل 50٪ بودن ضریب توجه فقط می توانیم بر روی 45 نیرو - روز حساب باز کنیم.

پس برآورد سرعت ما 45 Story Point به ازای هر اسپرینت می باشد. اگر هر داستان 5 روز برآورد شده باشد، پس ما در هر اسپرینت به طور حدودی 9 داستان را می توانیم انجام دهیم.

قرار دادن همه چیز در قالب طرح Release

تاکنون ما داستان ها را برآورد کردیم و هم سرعت تیم (45) را محاسبه کردیم. زمان این فرا رسیده است که بک لاگ محصول را در قالب اسپرینت جداسازی کنیم:

اهمیت	نام	برآورد
Sprint 1		
130	banana	12
120	apple	9
115	orange	20
Sprint 2		
110	guava	8
100	pear	20
95	raisin	12
Sprint 3		
80	peanut	10
70	donut	8
60	onion	10
40	grapefruit	14
Sprint 4		
35	papaya	4
10	blueberry	
10	peach	

هر اسپرینت دارای تعدادی داستان بر اساس ظرفیت و سرعت تیم می باشد (45).

قابل مشاهده است که برای انجام موارد "بایستی" و "باید" حداقل 3 اسپرینت لازم می باشد.

3 اسپرینت = 9 هفته = 2 ماه. آیا این همان ضرب العجلی است که ما به مشتری اعلام می کنیم؟ این وابسته به طبیعت خود قرار داد می باشد. ما معمولاً یک زمان برای مقابله با برآورد های بد، داستان های پیش بینی نشده، مشکلات پیش بینی نشده و غیره به این زمان اضافه می کنیم. در این مورد ما 1 ماه به عنوان رزرو در نظر می گیریم و تاریخ ارائه نسخه یک را 3 ماه اعلام می کنیم.

مسئله خوب اینجا است که هر 3 هفته می توانیم یک دمو حاضر کنیم و از مشتری دعوت نماییم تا محصول آماده شده را ببینید و در صورت نیاز قسمت هایی را تغییر بدهد (البته این کاملاً وابسته به نوع بسته شدن قرار داد می باشد).

سازگارسازی طرح Release

بعد از هر اسپرینت ما به سرعت واقعی تیم نگاه می کنیم. اگر سرعت واقعی با برآورد شده بسیار متفاوت باشد باید در سرعت برآورد شده تجدید نظر کنیم و برای اسپرینت های بعدی طرح Release تغییر و به روزرسانی شود. اگر این تغییر شما را به دردمر می اندازد، مالک محصول می تواند در این مورد با مشتریان مذاکره کند و یا او می تواند دامنه داستان ها را بدون شکستن قرار داد کاهش دهد. یا شاید بهتر باشد که تیم در سرعت خود تجدید نظر نماید و با از بین بردن موانع و یا تقویت ضریب توجه خود، سرعت خود را بالا ببرد.

مالک محصول می تواند با مشتریان بدین گونه مذاکره کند "سلام، ما طبق برنامه ریزی تعیین شده در حرکت هستیم و اعتقاد داریم خواهیم توانست که به هدف نهایی دست یابیم ولی فقط یک داستان بزرگ به نام "Embedded Pacman" وجود دارد که ساخت آن خیلی طول خواهد کشید. اگر اجازه بدهید آن را ما از لیست بایدهای نسخه یک حذف کنیم و بعد از ارائه نسخه اول بلافاصله این داستان را طی 3 هفته ارائه دهیم".

شاید خبر خوبی نباشد که معمولا مشتری قبول نمی کند ولی حداقل این خوب است که ما قبلا یک انتخاب در اختیار مشتری قرار دادیم. یا باید ما داستان های خیلی مهم را به موقع تحویل دهیم و یا هیچ کدام را تحویل ندهیم - کدامیک بهتر است - انتخاب با مشتری خواهد بود.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

13

چگونه XP را با اسکرام ترکیب می کنیم

اسکرام متدی است که بر روی مدیریت و سازماندهی پروژه و محصول متمرکز می باشد و XP متدی است که بر روی اصول برنامه نویسی واقعی بنا شده است. این همان دلیلی است که آن ها می توانند با هم ترکیب و مکمل هم شوند.

بدین وسیله بر طبق اسناد، شواهد و تجربیات با صدائی بلند می گویم که روش های اسکرام و XP به صورت مفیدی قابل ترکیب می باشند.

قصد دارم تعدادی از اصول با ارزش XP که قابل پیاده سازی در محیط اسکرام می باشد را معرفی نمایم. البته بیشتر تیم های ما از ترکیب این دو با هم استفاده نکرده اند ولی ما تجربه موفق در بعضی از تیم ها در مورد ترکیب اسکرام/XP داشته ایم. بعضی از اصول XP دقیقاً با اسکرام هم پوشانی دارند مانند "تیم"، "نشستن تیم باهم"، "داستان ها" و "بازی برنامه ریزی".

برنامه نویسی جفتی^{۲۶}

این مورد را در یکی از تیم های خود انجام دادیم و بسیار خوب عمل کرد. اما تیم های دیگر هم حداقل یک بار این را تجربه داشته اند.

بعضی از نتایج به دست آمده از برنامه نویسی جفتی:

- برنامه نویسی جفتی باعث بهبود کیفیت کدها می شود.
- برنامه نویسی جفتی ضریب توجه تیم را بهبود می دهد (زمانی را در نظر بگیرید که کسی پشت سر شما ایستاده است و می گوید "آیا این واقعا برای اسپرینت مورد نیاز می باشد؟").

²⁶ Pair Programming

- به طور غیر قابل باوری توسعه گرانی که سر سخته با این روش مخالف بودند بعد از مشاهده یادگیری سریع در این روش طرفدار پر و پا قرص این روش شدند.
- برنامه نویسی جفتی نباید هر روزه و دائما انجام شود.
- جابجایی جفت ها به طور معمول خوب است.
- برنامه نویسی جفتی بهبود دهنده فرآیند انتشار دانش در سطح گروه می باشد. به صورت خیلی سریع.
- بعضی از برنامه نویسان با جفت بودن راحت نیستند و به خاطر اینکه آنها با این روش راحت نیستند یک برنامه نویس خوب را کنار نگذارید.
- مرور کدها یک روش دوم و یا شق دوم روش برنامه نویسی جفتی می باشد.
- هدایت گر و یا Navigator (کسی که صفحه کلید را در اختیار ندارد) باید در جلوی خود یک کامپیوتر داشته باشد تا در صورت نیاز بتواند کدهای در ذهن خود را امتحان کند ، به مستندات رجوع کند و ...
- افراد را مجبور به برنامه نویسی جفتی نکنید. محیط و ابزار لازم را در اختیار آنها قرار دهید تا آنها به میل خود جفت شوند.

توسعه آزمایش محور (TDD^{۲۷})

این روش برای من مهمتر از هر دو متد اسکرام و XP می باشد. شما می توانید خانه، تلویزیون، سگ من را بگیرید، اما سعی نکنید که من را از انجام TDD منصرف کنید. اگر شما به TDD علاقه ندارید ساخت محصولاتان را به من واگذار نکنید زیرا من دزدکی از این روش استفاده خواهم کرد ...

خلاصه 10 ثانیه ای از TDD

توسعه آزمایش محور به این معنی می باشد که شما ابتدا یک تست اتوماتیک می نویسید. سپس به مقدار کافی کد می نویسید تا تست مربوطه پاس شود. بعدا برای بالا بردن خوانایی کد و حذف کدهای تکراری آن را رفاکتور می نمایید و این کار ادامه پیدا می کند.

بازتاب هایی از توسعه آزمایش محور

²⁷ Test-Driven Development

- TDD سخت است و این زمان خواهد برد که برنامه نویسی به این روش دست یابد. در بسیاری از موارد این مهم نیست که شما چقدر دوره دیده باشید و یا در چه کلاس هایی شرکت کرده باشید ، انجام TDD سخت است. بهترین راه حل برنامه نویسی جفتی با کسی است که TDD را خوب گرفته است. در این روش بهتر و زودتر می توان به این روش دست یافت.
- TDD تاثیر مثبت عمیقی در طراحی سیستم دارد.
- برای محصول های جدید زمان بر خواهد بود که بتوانید روش TDD را نهادینه سازی بکنید بخصوص تستهای یکپارچه جعبه-سیاه²⁸ ، ولی در عوض برگشت سرمایه سریع تر انجام خواهد شد.
- مطمئن شوید که زمان کافی را برای ایجاد سهولت در نوشتن تست ها صرف کرده اید. به این صورت که شما ابزار لازم را فراهم کرده اید ، کلاس های پایه و یا کمکی را آماده سازی کرده اید ، افراد را آموزش داده اید و امثالهم .

ما از ابزار زیر برای توسعه آزمایش محور استفاده می کنیم :

- jUnit / httpUnit / jWebUnit / TestNG / Selenium
- HSQLDB به عنوان یک دیتابیس درون حافظه ای جهت اهداف تست.
- Jetty به عنوان یک وب کانتینر مقیم در حافظه جهت اهداف تست.
- Cobertura
- Spring Frameworks برای نوشتن انواع مختلفی از تست ها (همراه با Mock ها ، بدون Mock ها ، با دیتابیس های خارجی ، با دیتابیس های مقیم در حافظه و غیره)

در محصولات پیچیده ما (از نظر TDD) تست های پذیرش جعبه سیاه را اتوماتیک می کنیم. این تست ها کل سیستم را در حافظه مقیم می کند و این بشدت در چرخه ساخت و اجرا تست ها تسریع ایجاد می کند.

علاوه بر این باعث ایجاد یک دلگرمی و اعتماد در توسعه گران می شود که آنها می توانند بدون نگرانی از خراب شدن قسمتی از برنامه اقدام به رفاکتور کردن کدها بکنند.

²⁸ Black-Box Integration tests

TDD در کدهای جدید

در تمام توسعه کدهای جدید از روش TDD استفاده می کنیم حتی برای قسمت های کم ارزش و یا پایه پروژه و هیچ عذر موجهی برای استفاده نکردن از این روش در نوشتن کدهای جدید قابل پذیرش نیست .

TDD در کدهای قدیمی

TDD سخت است اما سعی کردن در جای دهی TDD در داخل کدهای قدیمی که بر این اساس نوشته شده اند واقعا خیلی سخت است. چرا؟ خوب اگر من بخواهم این مورد را توضیح بدهم باید به اندازه این کتاب یک کتاب جدید بنویسم و این مورد را به آن کتاب وا می گذارم.

ما زمان بسیار زیادی برای ایجاد تست های اتوماتیک یکپارچه صرف می کنیم. اما حتی با وجود این تست های اتوماتیک یکپارچه , قسمت های مختلفی از سیستم می ماند که درون تست ها قرار نمی گیرد یعنی آنها زیر پوشش تست اتوماتیک قرار نمی گیرند. این تست ها فقط شامل قسمت کد می باشند و قسمت های غیر کد از این قاعده مستثنی می شوند.

برای هر Release یا ارئه ای از سیستم ما مجبور هستیم توسط یک تیم آزمایشگر , تعدادی آزمایش پیچیده بسرفت و پرفرنس را به صورت دستی انجام دهیم. تست های بسرفت بیشتر به صورتی دستی انجام می شوند و این به طور قابل ملاحظه ای سرعت توسعه و ارائه را پایین خواهد آورد. پس هدف اصلی این است که تمام تست ها خودکار شوند. بعد از مدتی کوباندن سرخودمان به دیوار نتوانستیم این موضوع را رفع نماییم.

بعد از مدتی تلاش برای اتوماتیک کردن این تست ها , به این روش رسیدیم که باید زمان تست های بسرفت را پایین بیاوریم. در آن زمان ما بر روی یک پروژه گیم کار می کردیم. معمولا برای انجام تست های بسرفت باید آزمایشگر ها منتظر می مانند یک تورنمنت بارگذاری شود تا آن ها بتوانند آزمایش ها را به صورت دستی انجام دهند. بنابراین با این روش ما ابزار ها و برنامه های کوچکی را برای پایین آوردن زمان تست های دستی مانند تست بسرفت ایجاد نمودیم.

این را اگر ما از اول انجام می دادیم شاید امکان اتوماتیک کردن آن ها هم به وجود می آمد و نوشتن تست بعد از کدها کاری است بسیار سخت و طاقت فرسا. قبل از نوشتن کد ها و طراحی سیستم باید در فکر تست های آن نیز باشیم.

طراحی Incremental

به این معنی می باشد که به جای اینکه از اول یک طراحی بسیار خوب ایجاد و آن را ثابت نگه دارید بهتر است که از ابتدا طراحی را ساده شروع کنید و به مرور آن را بهبود دهید.

ما معمولا زمان زیادی بر روی رفاکتور کردن و بهبود طراحی موجود صرف می کنیم و به ندرت وقت خود را صرف طراحی بزرگ UP-front یا طراحی کامل جز به جز می کنیم. درست است که گاهی با این روش به درد سر می افتیم (مثلا یک طراحی کوچک برای یک مورد در نظر می گیریم و بعدا متوجه می شویم که آن مورد خودش یک پروژه بزرگ بوده است) ولی در کل این روش طراحی مورد رضایت ما می باشد.

بهبود مداوم طراحی یکی از اثرات جانبی روش توسعه آزمایش محور می باشد.

یکپارچه سازی مداوم^{۲۹}

بیشتر محصول های ما دارای سیستم یکپارچه سازی بر اساس Maven یا QuickBuild می باشند. که این بسیار با ارزش می باشد و باعث صرفه جویی در وقت می شود. این بهترین راه حل برای جلوگیری از صحبت هایی مانند "این در سیستم من کار می کند ولی چرا برای تو Error می دهد؟!" می باشد. سرور ساخت مداوم همانند یک قاضی عمل می کند و گناهکاران را سریع معرفی می نماید. هر زمان که کدی توسط برنامه نویس به مخزن ارسال می شود، سرور ساخت فعال می شود و ابتدا پروژه را بر روی یک سیستم مشترک Build می کند بعدا تمام تست ها را از اول آزمایش می نماید. اگر موردی درست انجام نشود سریعاً ایمیلی به تمام اعضای تیم ارسال می شود که چیزی درست انجام نشده است و جایی از پروژه تخریب شده است. در ایمیل ارسالی ناحیه خطا نیز مشخص می شود.

²⁹ Continues integration

هر شب سرور Build متوالی تمام محصول را از ابتدا پابلیش و تست می کند و نتایج هر ساخت را به صورت گزارش ارائه می دهد.

با تنظیم سرور ساخت متوالی می توان کارهای زیادی انجام داد ولی آن های همیشه لازم نیستند.

مالکیت جمعی کد

مالکیت جمعی کد از طرف ما بسیار توصیه می شود ولی در همه تیم ها قابل دست یابی نیست. به این نتیجه رسیده ایم که در تیم های برنامه نویسی جفتی حس مالکیت جمعی کد ها بسیار بالاتر می باشد.

تیم هایی با حس مالکیت جمعی کد بالا بسیار قوی و نیرومند خواهند بود برای مثال اگر فردی کلیدی از یک تیم با حس مالکیت جمعی کد بالا در مابین اسپرینت مرخص شود ، آن اسپرینت به دلیل نیرومند بودن تیم ، نخواهد مرد.

محل کار آگاهی بخش

محل کار تیم ها باید آگاهی دهنده باشد. همه تیم ها در اتاق خود وایت برد و تابلوی وظایف دارند ولی فقط تعدادی از آنها به بهترین نحو استفاده می کنند. اگر شما به اتاق این تیم ها نگاه کنید در همه جا اطلاعات سودمندی در مورد پروژه ، محصول ، طراحی و غیره خواهید یافت. بعضی از تیم ها از بهترین ابزار آگاهی دهنده یعنی همان تابلوی وظایف نیز استفاده نمی کنند.

استاندارد کدنویسی

ما بالاخره شروع به تعریف استانداردهای کدنویسی کردیم که بسیار سودمند می باشند. و انجام این کار زیاد طول نمی کشد. این کار را با کمی استاندارد ساده شروع کردیم و به مرور استانداردهای خود را بهبود دادیم.

بعضی از برنامه نویسان برای خود استایل کدنویسی مشخصی دارند. جزئیات کوچکی مانند چگونه باید Exception ها هندل شوند ، چگونگی کامنت کدها ، زمانیکه آنها مقدار Null برمی گردانند و غیره. در بعضی از موارد تفاوت ها مهم نیستند ولی در بعضی از موارد ممکن است این منجر به غیر همسان

سازی طراحی و یا سخت شدن خوانایی کدها شود. در این حالت استاندارد کدنویسی با ارزش می باشد پس باید در استاندارد به مواردی که برای شما اهمیت دارند پردازید.

نمونه ای از استانداردهای کدنویسی ما :

- هر زمان که خواستید یکی از این قوانین را بشکنید دقت کنید که اولاً دلیل خوبی داشته باشید ثانیاً اینکه آن را مستند نمایید.
- از محاوره کد Sun استفاده شود .
- هرگز Exception ها را بدون مقدار پشته Trace لاگ نکنید.
- از روش Dependencey Injection برای ایجاد کلاس های نا وابسته استفاده کنید. (به جز مواردی که امکان ندارند)
- از کوتاه سازی های (مخفف سازی) بی مورد پرهیزید. مثلاً DAO نوع خوب کوتاه سازی می باشد.
- متدهایی که خروجی آنها آرایه ها می باشند نباید مقدار Null برگردانند به جای آن یک آرایه خالی برگردانند.

گام های ثابت / کار تقویت شده

بسیاری از کتاب توسعه نرم افزار Agile ادعا می کنند که اضافه کار باعث کاهش بهره وری در توسعه نرم افزار می شود.

در حدود یک سال قبل ما یک تیم داشتیم که به صورت وحشتناک اضافه کاری انجام می داد. محصولی که آنها در حال انجام آن بودند بسیار از نظر کیفیت پایین آمده بود به همین دلیل مشتریان محصول عصبانی بودند و دائماً آنها را تحدید می کردند و این باعث می شد که تیم برای رفع مشکلات , بیشتر اضافه کاری نماید. با این همه اضافه کاری تیم فرصت این را نداشت تا کیفیت محصول را ارتقاء دهد.

بعد از چند ماه , تیم را به حالت اول خود بازگردانیم و تیم در ساعات کاری معمولی خود کار می کرد. به طور غیر قابل باوری , بهره وری نیروی انسانی و کیفیت محصول بسیار بالا رفت.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

14

چگونه تست می کنیم

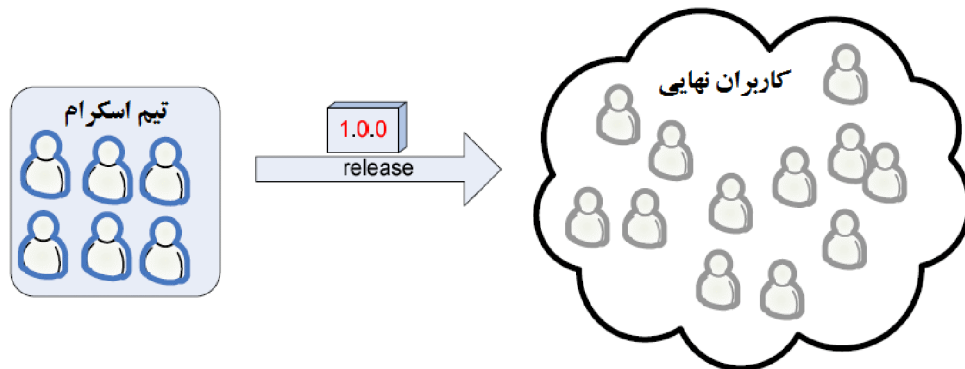
فاز تست کمی سخت می باشد. من مطمئن نیستم که این بخش در اسکرام سخت می باشد و یا کلا این بخش در توسعه نرم افزار سخت می باشد.

آزمایش کردن معمولا یکی از بخش هایی است که در سازمان های مختلف متنوع انجام می شود. اما تست و ایجاد تنوع می تواند به مواردی وابسته باشد: تعداد آزمایشگران، چقدر بودن میزان خودکاری سازی تست ها، نوع سیستم های در حال توسعه، اندازه چرخه Release ها و غیره.

ما تجربیات زیادی در مورد تست در محیط اسکرام کسب نموده ایم و در این بخش می خواهیم به معرفی روش های مختلف تست در محیط اسکرام پردازم.

شما احتمالا نمی توانید فاز تست پذیرش³⁰ را کنار بگذارید

در محیط ایده آل اسکرام نتیجه هر اسپرینت نسخه قابل ارائه از سیستم می باشد. اما آیا فقط Deploy کردن کافی است؟



اشتباه

³⁰ Acceptance Test

تجربه این را می گوید که این روش معمولاً کارا نیست. باگ های بزرگ و زننده ای وجود خواهند داشت. اگر کیفیت در اولویت های شما باشد، انجام تست پذیرش به صورت دستی نیاز می باشد. البته زمانی که آزمایش گرهایی به صورت اختصاصی برای چکش کاری سیستم همراه تیم نیستند و یا زمان کافی برای تست وجود ندارد و یا سخت افزار مورد نظر برای تست موجود نمی باشد.

در صورت وجود آزمایشگر در تیم، آنها نیز به همان صورت کاربران نهایی به سیستم دست رسی خواهند داشت. بدین معنی که بیشتر موارد باید تکمیل شده باشند.



تیم آزمایش باگ های موجود در نسخه ارائه شده را کشف می کند و تیم اسکرام باگ ها را رفع و نسخه 1.0.1 را ارائه می دهد. شما می توانید نسخه رفع باگ شده یعنی 1.0.1 را به کاربران نهایی تحویل دهید.

زمانی که من گفتم تست پذیرش، منظور من کل دوره آزمایش، باگ زدایی و ارائه نسخه جدید و کلاً تا زمانی که یک نسخه خوب قابل ارائه به وجود آید، می باشد.

به حداقل رساندن فاز تست پذیرش

فاز تست پذیرش آسیب رساننده است. به همین خاطر حس می شود که این یک روش غیر Agile می باشد. ولی با این حال امکان کنار گذاشتن آن وجود ندارد ولی می توانیم این فاز را به حداقل ممکن برسانیم. بخصوص می توان تعداد زمان های تست پذیرش را به حداقل رساند. بدینگونه که:

- کیفیت کد ارائه شده از طرف تیم اسکرام را به حداکثر برسانید.

- تاثیر پذیری کار تست دستی را به حداکثر برسانید (برای مثال بهترین آزمایش گرها را برای تست پیدا کنید، بهترین ابزارها را به آنها بدهید، ...)

چگونه ما کیفیت کد ارائه شده از طرف تیم اسکرام را به حداکثر می‌رسانیم؟ خوب، راه‌های زیادی وجود دارند که این دو روش آنهایی هستند که خوب جواب داده‌اند:

- قرار دادن آزمایشگرها در داخل تیم‌های اسکرام
- کم‌تر انجام دادن داستان‌ها به ازای اسپرینت

افزایش کیفیت با قرار دادن آزمایشگرها داخل تیم‌های اسکرام



منظور من از آزمایشگر داخل تیم‌ها یک شخصی است که مهارت پایه او تست کردن می‌باشد و نه شخصی که فقط نقش او تست کردن می‌باشد. یعنی مهارت کد نویسی هم باید داشته باشد.

توسعه دهندگان معمولاً آزمایشگران نفرت‌انگیزی می‌باشند. بخصوص توسعه دهندگانی که کدهای خود را تست کنند.

آزمایشگر " نشانه گذار پایان " است

آزمایشگری در درون تیم‌های اسکرام یک شغل مهم به حساب می‌آید. او کسی است که مهر نهایی شدن را خواهد زد. هیچ موردی در اسپرینت تکمیل شده به حساب نخواهد آمد مگر اینکه آزمایشگر بگوید. من مواردی را دیده‌ام که توسعه‌گران گفته‌اند آن تکمیل شده است در حالی که واقعا تکمیل

نشده بود. حتی اگر شما تعریف دقیق و واضحی از Done و یا تکمیل شدن داشته باشید، توسعه گران معمولاً آن را فراموش خواهند کرد. برنامه نویسان ما معمولاً آدم های بی صبری می باشند که هر چه زودتر می خواهند به موارد بعدی بروند.

آقای T (تست کننده) چگونه خواهد فهمید که موردی کامل انجام شده است؟ خوب، قبل از هر چیزی باید آن را تست بکند. در بعضی از موارد در هنگام شروع تست، آن مورد به توسعه گر بر گردانده می شود زیرا حتی آیتم قابل تست کردن نمی باشد. زمانی که آقای T در حال تست ویژگی می باشد باید ویژگی را از مابین قوانین تعریف Done بگذراند و البته در حضور خود توسعه گر. در هر مرحله از آزمایش که مشکلی به وجود بیاید توسعه دهنده آن را مشاهده خواهد کرد و مورد را مانند تعریف Done تکمیل خواهد کرد.

اثر جانبی وجود آزمایشگر در داخل تیم این می باشد که تیم شخصی را در اختیار دارد که او به طور کامل در حال آماده سازی دموی اسپرینت می باشد.

زمانی که آیتمی برای آزمایش نباشد آزمایشگر چه خواهد کرد؟

این سوالی است که معمولاً آقای T از مدیر اسکرام می پرسد "مدیر اسکرام، آیتمی برای تست کردن وجود ندارد، من چه کاری باید انجام دهم؟". این معمولاً چند هفته قبل از پایان اولین داستان حادث می شود. خوب تست کننده باید در طی این مدت چه کاری انجام دهد؟

خوب، قبل از همه او باید آماده تست کردن شود. با نوشتن تست های ویژه، آماده سازی محیط تست و غیره و زمانی که اولین مورد آماده شد، بدون انتظار باید آزمایشگر درون ویژگی شیرجه بزند.

اگر تیم به روش TDD کار می کند از اولین روز زمانی را برای نوشتن تست ها صرف می کنند. آقای T باید با توسعه گرانی که تست می نویسند جفت شود. اما در جفت شدن او باید نقش هدایت گر را بازی نماید و صفحه کلید در دست خود توسعه گر باقی بماند. یک آزمایش گر خوب معمولاً با خود چندین نوع آزمایش را به همراه خواهد داشت و در این جفت سازی، اطلاعات خوبی رد و بدل می شود و تست های خوبی نوشته می شود.

اگر تیم TDD کار نمی کند و یا به اندازه کافی تست نوشته نمی شود، آزمایشگر می تواند کار دیگری که با انجام آن به تیم در رسیدن به هدف اسپرینت کمک می کند انجام دهد. مثلا او می تواند مانند بقیه شروع به کد نویسی بکند. اگر نتواند برنامه نویسی کند می تواند در مسائل غیر برنامه نویسی فعال شود.

مثلا شکستن داستان ها به وظایف می تواند یکی از این کارها باشد که تیم در داخل برنامه ریزی اسپرینت انجام می دهد. البته کارهای زیاد غیر برنامه نویسی در اسپرینت وجود دارند که باید در اسپرینت تکمیل شوند. آقای T می تواند در تشخیص و انجام این کارها به تیم کمک کند، حتی اگر او نتواند برنامه نویسی بکند.

مثال هایی از موارد غیر برنامه نویسی که در اسپرینت باید انجام شوند:

- راه اندازی محیط تست.
- آشکار سازی و واضح سازی نیازمندی ها.
- بحث و آشکار ساختن جزئیات Deploy.
- نوشتن مستندات Deployment (یادداشت های ارائه، RFC و یا هر چیز دیگری که سازمان شما انجام می دهد).
- ارتباط با منابع خارجی (مانند طراحان رابط کاربری).
- بهبود اسکریپت ساخت.
- شکست دادن بیشتر و بهتر داستان ها به وظایف.
- تشخیص سوالات اساسی توسعه گران و جواب دادن به آنها.

در حالت برعکس، زمانی که آقای T در تنگنا قرار گرفته باش ما چه کاری باید انجام دهیم؟ فرض کنید در آخرین روز اسپرینت می باشیم و همه چیز یکباره تکمیل شده است ولی آقای T فرصت آزمایش همه موارد را ندارد. ما چه کاری انجام می دهیم؟ خوب ما معمولا همه اعضای تیم را به عنوان دست یار آقای T در نظر می گیریم. او به آنها می گوید که چه آزمایش هایی برای گارانتی ویژگی نیاز است که انجام شود، و این اختیار را به تیم می دهد که آنها تایید تکمیل شدن برخی از ویژگی ها را بزنند.

بلی ، آقای T یک نقش اختصاصی در تیم اسکرام می باشد ولی با این حال به او اجازه انجام کارهای دیگری هم داده شده است.

افزایش کیفیت با انجام دادن کار کم طی اسپرینت

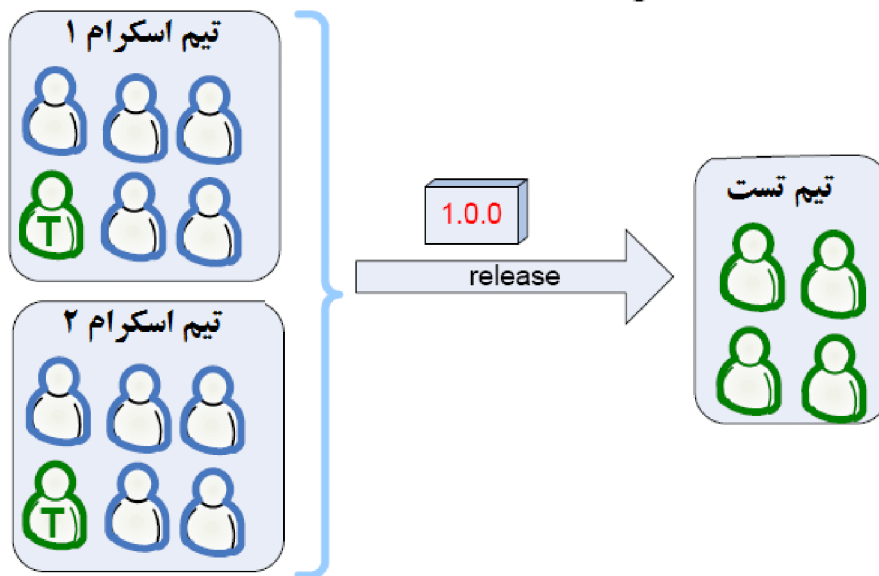
این مورد به جلسه برنامه ریزی اسپرینت بر می گردد. برای انجام این اصل می توانید میزان قرار دادن ویژگی در اسپرینت را در جلسه برنامه ریزی اسپرینت کاهش دهید. اگر شما مشکل کیفیت دارید و یا در چرخه انجام تست پذیرش با مشکل مواجه می شوید می توانید به سادگی تعداد موارد اسپرینت را کاهش دهید. این معمولا به طور خودکار باعث افزایش کیفیت ، کاهش تعداد باگ ها ، کاهش چرخه تست پذیرش و غیره خواهد شد. کاهش باگ می تواند به تیم کمک کند که به جای صرف زمان خود برای رفع باگ ها ، موارد جدید را زودتر انجام بدهند.

این همیشه خوب است که کمتر بسازید ولی پایدار بسازید به جای اینکه بیشتر بسازید ولی مجبور شوید که از اول بسازید.

آیا تست پذیرش باید جزوی از هر اسپرینت باشد؟

ما در این بحث بسیار نوسان داشتیم. بعضی از تیم های ما فاز تست پذیرش را در داخل اسپرینت انجام می دادند ولی بیشتر تیم های ما برعکس عمل می کردند: به چند دلیل:

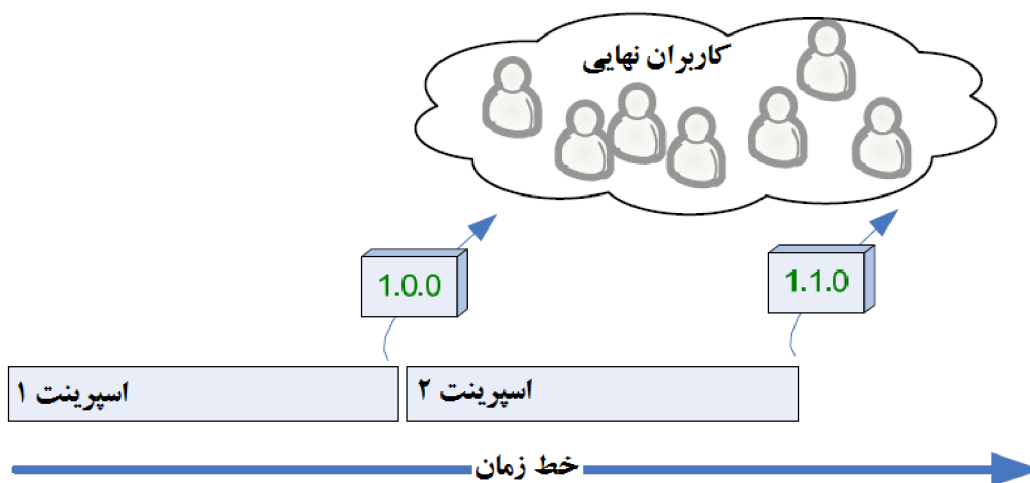
- اسپرینت یک زمان مشخص دارد. ولی انجام تست پذیرش در طی یک زمان مشخص خیلی سخت می باشد. اگر زمان اسپرینت تمام می شد و شما هنوز باگ های حیاتی داشتید که هنوز رفع نشده اند ، چه کار خواهید کرد؟ آیا ارائه را همراه باگ های باقی مانده انجام خواهید داد؟ آیا تا اسپرینت بعدی صبر خواهید کرد؟ پس ما به همین خاطر تست های دستی را به بیرون از اسپرینت انتقال دادیم.
- اگر تیم های متفاوتی بر روی یک محصول کار کنند ، یکی از آنها تست ها را زودتر انجام خواهد داد و یکی هنوز بعد از مدتی به تست ادامه خواهد داد که زمانبندی اسپرینت ها در بین تیم ها به هم خواهد خورد. در این حالت همه باید منتظر تیم های بعدی بمانند تا بتوانند یک نسخه از سیستم را ارائه کنند.



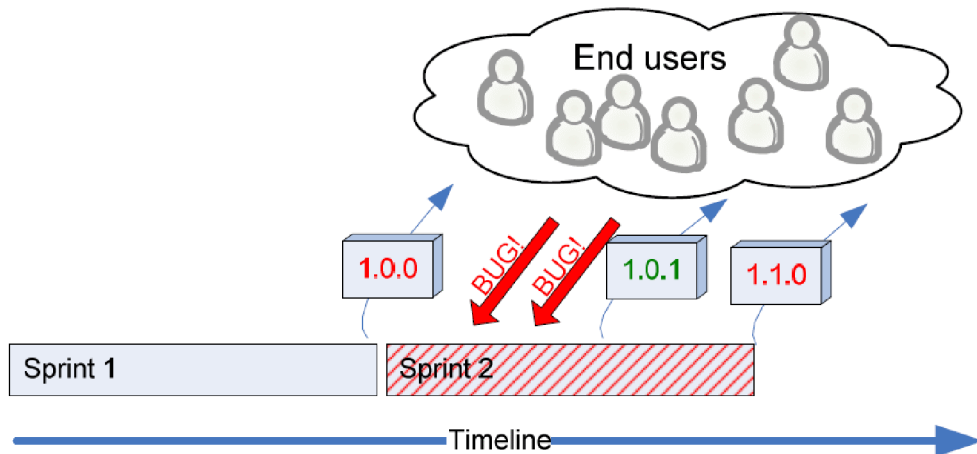
این به این معنی نمی باشد که این روش خوبی است ولی باید برای ارائه چند تیمی اینگونه عمل شود.

چرخه اسپرینت در مقابل چرخه تست پذیرش

در یک محیط ایده آل و کامل اسکرام شما تا زمانی که تیم ها یک تولید قابل ارائه از محصول را آماده می کنند نیازی به فاز تست پذیرش ندارید و می توانید خروجی اسپرینت را مستقیماً به مشتری ارائه کنید.



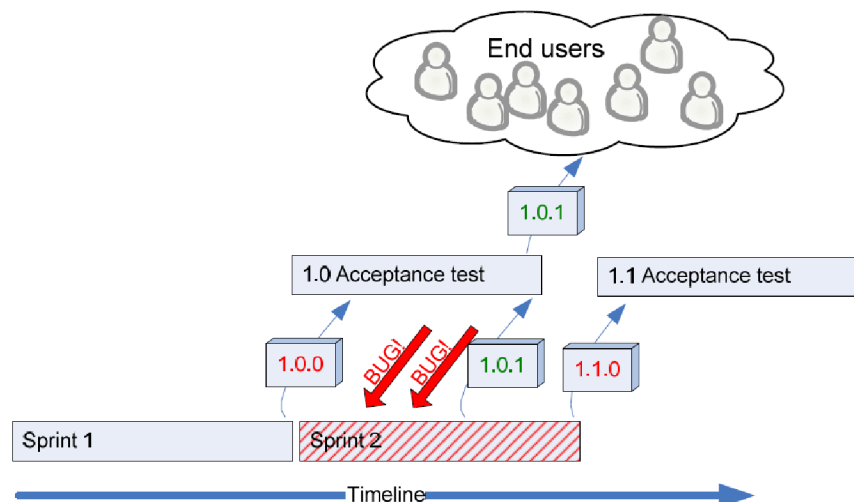
خوب عکس پایین کمی واقعی تر می باشد:



بعد از اسپرینت 1، نسخه باگ دار 1.0.0 ارائه می شود. در اسپرینت دوم فرآیند گزارش باگ ها شروع می شود و تیم شروع به باگ زدایی می کند و آنها مجبور می شوند در نیمه اسپرینت دوم نسخه باگ زدایی شده نسخه قبلی را با شماره 1.0.1 ارائه کنند. در انتهای اسپرینت دوم، یک سری ویژگی جدید طی نسخه 1.1.0 ارائه خواهد شد که کاملاً واضح می باشد این نسخه از نسخه قبلی هم پر باگ تر خواهد بود.

قسمت هاشور زده قرمز در اسپرینت دوم حاکی از آشفتگی در این اسپرینت می باشد.

خیلی روش خوبی نبود؟! خبر بد این است که این مشکل حتی اگر شما تیم تست پذیرش هم داشته باشید باز باقی خواهد ماند. فقط فرقی که خواهد داشت این می تواند باشد که اولاً گزارش های بیشتر و بهتری در مورد باگ ها دریافت خواهید کرد و ثانياً کاربران را عصبانی نخواهید کرد که ممکن است به تجارت این محصول لطمه وارد شود.



هیچ راه حل ساده ای برای این مشکل پیدا نکردیم و ما از مدل های مختلفی برای حل این مشکل استفاده می کنیم.

قبل از همه ، افزایش و به حداکثر رسانی کیفیت کدهای تیم اسکرام می باشد. هزینه پیدا کردن و رفع خطاها در داخل اسپرینت شدیداً پایین تر از پیدا کردن آنها بعد از اسپرینت خواهد بود.

با این حال مشکل پا برجا خواهد ماند زیرا ما با این روش ها فقط تعداد باگ های داخل محصول ارائه شده را کاهش خواهیم داد و دوباره شاهد گزارش های باگ خواهیم بود. برای مقابله با این چکار می کنیم؟

روش 1: "تا زمانیکه آیتم قبلی در حال تولید می باشد آیتم جدیدی را شروع نکنید."

به نظر خوب می رسد؟ اینطور نیست!

ما در بعضی از اوقات و نه همیشه از این روش در چندین تیم استفاده کردیم. این روش به این صورت می باشد که بعد از هر اسپرینت یک چرخه ارائه البته نه با یک زمان مشخص شروع می شود که در این چرخه کلیه کارهای مربوط به تولید و ارائه انجام می شوند و تا زمانیکه این چرخه به درستی تکمیل نشود اسپرینت بعدی شروع نمی شود.



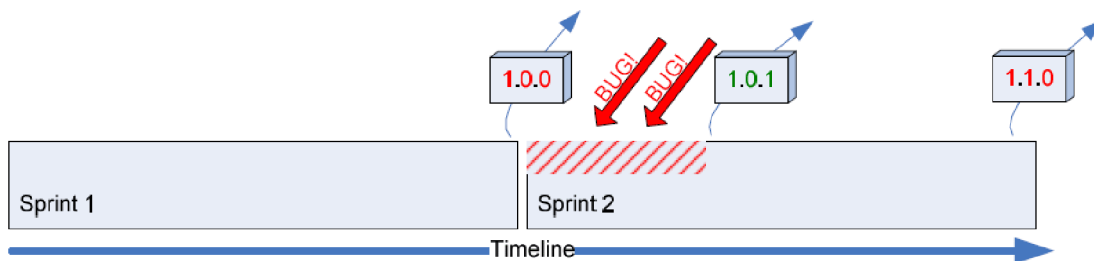
ولی این روش باعث به هم خوردن توالی ارائه دموها خواهد شد و حتی می تواند در قرار دادهای بسته مشکل به وجود آورد. دیگر در این روش نمی توان گفت که هر 3 هفته یک دمو قابل ارائه خواهیم داشت.

روش 2: "می توانید ساخت آیتم های جدید را شروع کنید ولی آیتم های مشکل دار اسپرینت قبلی نیز برای تولید در این اسپرینت باید اولویت بندی شوند"

این در واقع فعلاً روش مورد علاقه ما می باشد.

به طور کلی زمانی که یک اسپرینت را تمام می کنیم ، اسپرینت بعدی را شروع می کنیم ولی با این دید که منتظر تولید و ارائه باگ ها و مشکلات اسپرینت قبلی هم هستیم. در این حالت باید طول و زمان اسپرینت کمی بزرگ تر باشد تا زمان کافی هم برای رفع خطاهای اسپرینت قبلی باشد و هم موارد جدید ساخته شوند.

خوشبختانه در طی اسپرینت ها تعداد باگ های اسپرینت قبلی برای باگ زدایی در اسپرینت بعدی کاهش می یابد و این باعث خواهد شد که تعداد افراد کم تری مشغول رفع باگ شوند و به این ترتیب آیتم های بیشتری ساخته خواهند شد.



برای ایجاد زمان جهت رفع باگ در برنامه ریزی اسپرینت ضریب توجه تیم را کمی پایین تر در نظر می گیرند تا تیم فرصت این را داشته باشد تا باگ های اسپرینت قبلی را رفع کند. البته در جلسات بعدی که تعداد باگ ها کاهش خواهد یافت می توان ضریب توجه را دوباره افزایش داد.

روش بد – "تمرکز برای ساخت موارد جدید"

این بدین معنی می باشد که ما "به جای در تولید قرار دادن آیتم های گذشته فقط بر روی ساخت موارد جدید تمرکز می کنیم". چه کسی این کار را می کند؟ خوب این اشتباه را ما در اوایل راه انجام می دادیم و مطمئن هستیم که شرکت های جدید هم این اشتباه را مرتکب خواهند شد.

این عارضه مربوطه به استرس می باشد. بیشتر مدیران این را واقعا درک نمی کنند ، زمانی که کدنویسی داستان ها تمام شد ، آنها انتظار دارند زود ارائه انجام شود. بعد از ارائه آنها انتظار دارند تا زمانیکه آیتمی موجود می باشد کار ساخت آیتم جدید متوقف نشود.

از حلقه اتصال کند خود پیشی نگیرید

در نظر بگیرید که تست پذیرش کندترین حلقه زنجیره شما می باشد. این کندی شاید به خاطر کم بودن تعداد آزمایشگران باشد و یا به خاطر پایین بودن کیفیت کدهای برنامه نویسان باشد.

در نظر بگیرید که تیم تست شما در بهترین حالت می توانند 3 ویژگی را هر هفته آزمایش کنند (این فقط یک مثال می باشد) و این را نیز در نظر بگیرید که تیم توسعه شما می توانند در هر هفته 6 ویژگی جدید را ایجاد کنند.

این می تواند باعث وسوسه مدیران و یا مالک محصول شود تا آنها تیم را برای تولید 6 ویژگی در هفته برنامه ریزی کنند.

این واقعا باعث صدمه به محصول خواهد شد.

به جای این سعی کنید برای هر هفته 3 ویژگی برنامه ریزی کنید تا بتوانید مشکلات ویژگی ها را کشف و رفع نمایید. برای مثال:

- داشتن چند توسعه گر که بتوانند به جای تست کننده نیز کار کنند.
- ابزاری که بتواند کار آزمایش را آسان تر کند.
- اضافه کردن تست های اتوماتیک بیشتر.
- طول اسپرینت را افزایش دهید و تست پذیرش را در داخل اسپرینت انجام دهید.
- اسپرینت هایی را با عنوان اسپرینت تست تعریف کنید تا تمام تیم در این اسپرینت به عنوان تست کننده فعالیت کنند.
- تست کنندگان بیشتری را استخدام کنید (حتی اگر این به این معنی باشد که باید توسعه دهندگانی حذف شوند).

ما تمام این روش ها را تجربه کرده ایم و به نظر من بهترین روش موارد 2 و 3 یعنی ابزار راحت کننده تست و اضافه کردن تست اتوماتیک بیشتر می باشد.

جلسات بازبینی بهترین مجمع برای مشخص کردن کندترین حلقه در زنجیره شما می باشد.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

15

ما چگونه چندین تیم اسکرام را اداره می کنیم

خیلی از مسائل با افزایش تعداد تیم های کارکننده بر روی یک محصول بسیار مشکل تر خواهند شد. البته این مسئله تنها برای اسکرام نیست و یک مسئله جهان شمول می باشد. توسعه گر بیشتر = پیچیدگی بیشتر .

مهمترین مسائلی که در تعداد افراد زیاد به وجود می آیند این است که :

- چند تیم باید ایجاد کنیم؟
- چگونه نفرات را بین تیم ها تقسیم کنیم؟

چند تیم باید ایجاد کنیم

چرا ما باید خودمان را به در دسر بیندازیم و به جای یک تیم چند تیم به وجود بیاوریم؟

بزرگترین تیم ما حدودا شامل 11 نفر می باشد. این ها با هم کار می کنند ولی نه خیلی خوب. معمولا جلسه روزانه اسکرام آنها بیشتر از 15 دقیقه به طول می کشد. اعضای تیم نمی دانند که دیگر اعضای تیم چه کاری انجام می دهند و این باعث به وجود آمدن سردرگمی در تیم می شود. این برای مدیر اسکرام نیز سخت خواهد بود تا تمام تیم را نسبت به هدف تراز نگه دارد و رفع مشکلات در این حالت بسیار سخت تر می باشد.

راه دیگر این است که این تیم به 2 تیم مجزا تقسیم شود. اما آیا وضعیت بهتر خواهد شد؟ نه لزوما.

اگر تیم تجربه استفاده از اسکرام را داشته باشد و با این متد راحت باشد و یک راه منطقی برای تقسیم هدف به دو شیار مجزا باشد که هر دو شیار شامل یک سورس کد نباشند ، می توانم بگویم که تقسیم

یک تیم به دو تیم ایده بسیار خوبی است. به عبارتی من بیشتر می پسندم با وجود مشکلات و معایب تیم بزرگ به همین روش یعنی یک تیم بزرگ بچسبم.

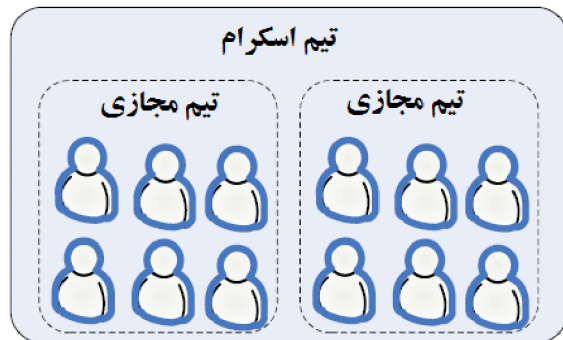
ایده تقسیم یک تیم بزرگ به تیم های کوچکتر زمانی موثر می باشد که تیم های کوچک به وجود آمده با هم تداخل نداشته باشند. اگر تیم ها تداخل داشته باشند ممکن است مشکلات آن بیشتر از یک تیم بزرگ شود.

تیم های مجازی

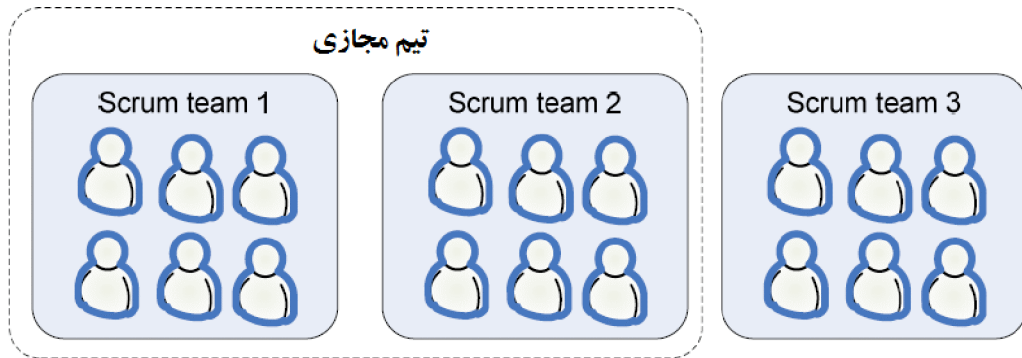
شما از کجا خواهید فهمید که تصمیم شما در مورد تیم بزرگ و یا چند تیم کوچک درست یا اشتباه بوده است؟

اگر چشم ها و گوش های خود را باز نگه دارید حتما متوجه شکل "تیم های مجازی" خواهید شد.

مثال 1: شما تصمیم گرفتید که یک تیم بزرگ داشته باشید. اما زمانیکه به تیم در طول اسپرینت نگاه می کنید که چه کسی با چه کسی صحبت می کند متوجه می شوید تیم به طور مجازی به دو تیم کوچک تقسیم شده است.



مثال 2: شما تصمیم می گیرید که 3 تیم مجزای کوچک داشته باشید. اما زمانیکه به تیم در طول اسپرینت نگاه می کنید که چه کسی با چه کسی صحبت می کند متوجه می شوید که تیم 1 دائما با تیم 2 در ارتباط می باشد و 3 به تنهایی کار می کند.



خوب معنی این چیست؟ استراتژی تقسیم تیم شما اشتباه بوده است؟ بلی و البته فقط زمانی که تیم های مجازی به صورت پایدار به نظر بیایند و نه به صورت موقت.

به مثال 1 دوباره نگاه کنید. اگر این حالت را مشاهده کردید که به صورت پایدار ادامه داشت مثلا تا آخر اسپرینت ادامه داشت شما می تواند در شروع اسپرینت جدید این تیم بزرگ را به دو تیم مجزا تقسیم کنید و این می تواند یک تصمیم درست باشد.

حالا به مثال 2 دوباره نگاه کنید. اگر ارتباط این تیم ها پایدار بود مثلا تا آخر اسپرینت ، می توانید تیم ها را به یک تیم بزرگ تبدیل کنید. اما باید مطمئن شوید که ارتباط آن ها موقت نیست. این را می توانید در جلسه بازبینی اسکرام مطرح نمایید و نظرات تیم ها را در این مورد جویا شوید.

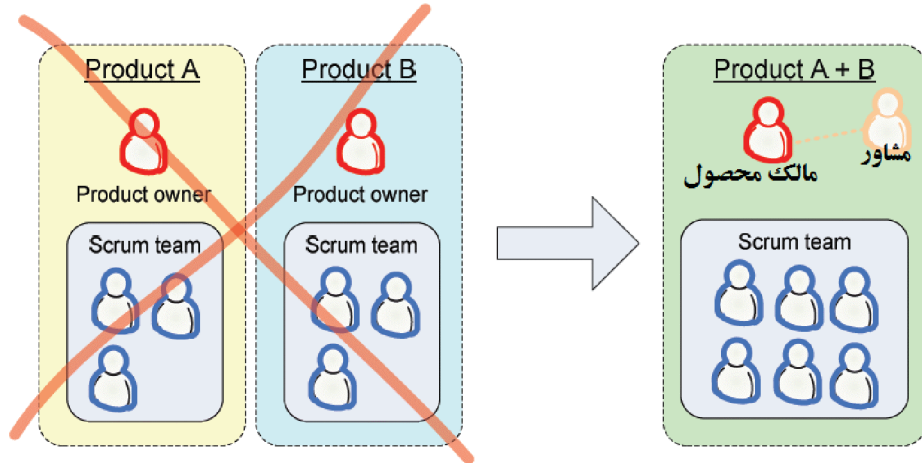
تقسیم تیم واقعا یکی از سخت ترین کارها در اسکرام می باشد. سعی کنید این تیم های مجازی را در داخل اعضا پیدا کنید و سعی کنید این مورد را در جلسه بازبینی با اعضا به نتیجه برسید. اما نگران تیم بندی خوب نباشید زیرا دیر و یا زود یک راه حل خوب پدید خواهد آمد. مسئله مهم این است که اعضا در تیم های خود راحت باشند.

بهترین اندازه تیم

در کتاب هایی که من خوانده ام آن ها ادعا داشتند که بهترین اندازه تیم "5-9" می باشد.

با توجه به تجربه ، بهترین اندازه تیم "3-8" نفر می باشد و من اعتقاد دارم که رسیدن به بهترین اندازه تیم کمی مشکل می باشد.

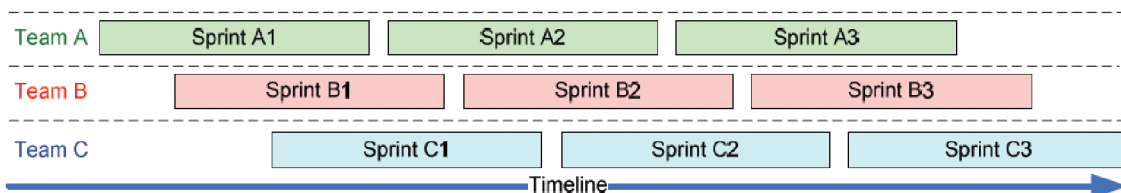
فرض کنید 2 محصول متفاوت دارید و برای هر محصول یک تیم 3 نفره در اختیار دارید. هر دوی آنها بسیار کند حرکت می کنند. این شاید ایده خوبی باشد که هر دو تیم ترکیب شوند و یک تیم 6 نفره را تشکیل بدهند و این تیم بزرگ مسئول هر دو محصول باشد. در این حالت به یکی از دو مالک محصول اجازه بدهید تا برود. (یا به او نقش مشاور پیشنهاد دهید)



فرض کنید شما یک تیم 12 نفره اسکرام دارید. به خاطر کدهای کثیف امکان به وجود آمدن دو تیم مجزا وجود ندارد. زیرا برای رفع این کدهای کثیف تداخل بین اعضای تیم ها به وجود خواهد آمد. برای جداسازی این تیم ها باید ابتدا تداخل ها حل شوند تا بتوان دو تیم مجزا به وجود آورد.

اسپرینت های همزمان آری یا نه ؟

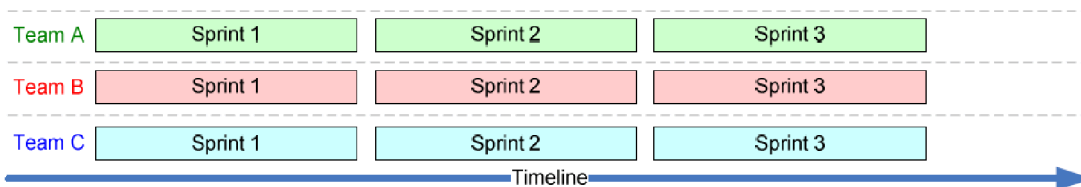
فرض کنید 3 تیم اسکرام داریم که همه آنها بر روی یک محصول کار می کنند. آیا آغاز و پایان اسپرینت های آنها باید همزمان باشد؟ اولین روش ما ایجاد هم پوشانی بین اسپرینت های متفاوت بود.



به نظر خوب می‌رسد. آنها اسپرینت‌ها را به صورت هم‌پوشانی آغاز و تمام می‌کنند. بدین صورت مالک محصول خواهد توانست در جلسات همه تیم‌های شرکت کند. هر هفته یک دمو ارائه خواهد شد. مبارک باشد.

بلی، می‌دانم که دلایل ارائه شده زیاد متقاعد کننده نیستند.

من این مسئله را با کن شوئبر (مدع اسکرام) در جایی مطرح کردم و او گفت که این ایده بسیار بدی است و اگر اسپرینت‌ها به صورت هم‌زمان آغاز و پایان شوند بسیار بهتر خواهد بود. من دلایل او را دقیقاً به خاطر ندارم ولی بعد از مدتی کاملاً متقاعد شدم.

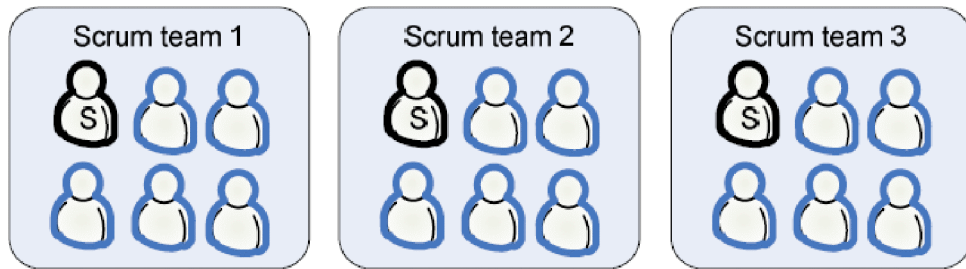


این راه حلی هست که ما انجام دادیم و هیچ وقت آن را ترک نخواهیم کرد. مزایای روش اسپرینت‌های هم‌زمان به این قرار می‌باشد:

- به طور طبیعی یک زمان مشخص برای دوباره چیدن نفرات بین اسپرینت‌ها به وجود می‌آید. اما در روش هم‌پوشانی چنین فرصتی فقط در آخر پروژه بدست خواهد آمد.
- همه تیم‌ها می‌توانند طی یک اسپرینت برای یک هدف کار کنند، جلسه برنامه ریزی اسپرینت را با هم برگزار کنند که این باعث به وجود آمدن همکاری بین تیم‌ها می‌شود.
- نیاز کمتر به هزینه مدیریت - جلسه برنامه ریزی کمتر - دموهای کمتر - و ارائه‌های کمتر.

چرا ما نقش "رهبر تیم" را معرفی کردیم

فرض کنید ما یک محصول به همراه 3 تیم داریم.



شخص با مارک P همان مالک محصول می باشد. اشخاص سیاه رنگ با مارک S مدیران اسکرام می باشند. در این مثال همه اشخاص در یک سطح قرار دارند.

با در نظر گرفتن این فرض، چه کسی تصمیم خواهد گرفت که چه کسی در کدام تیم باید باشد؟ مالک محصول؟! هر سه مدیر اسکرام با هم؟ یا هر کس خودش انتخاب خواهد کرد که در کدام تیم باشد؟ اما اگر همه بخواهند که در تیم 1 باشند چگونه خواهد شد (بدلیل اینکه مدیر اسکرام این تیم خوب به نظر می رسد).

اگر بعدا به نظر برسد که وجود تیم 3 نیاز نیست و بهتر است که 3 تیم به 2 تیم تبدیل شوند پس باید یکی از مدیران اسکرام کنار گذاشته شود. کدام یک کنار گذاشته خواهد شد؟ این تصمیم را چه کسی خواهد گرفت؟

در بعضی از شرکت ها این یک موضوع حساس می باشد.

به نظر می رسد که باید به مالک محصول اجازه عزل و نصب افراد در تیم ها داده شود ولی واقعا این حق مالک محصول می باشد؟ مالک محصول فقط متخصص دامنه پروژه می باشد و او این حق را ندارد که در مسائل داخلی و جزئیات تیم ها دخالت کند. بخصوص که "جوجه" می باشد (اگر اطلاع جوجه و خوک را نشنیده اید می توانید در گوگل کلید واژه Chicken And Pig را جستجو کنید).

این مسئله را با معرفی نقشه "رهبر تیم" حل نموده ایم. این اصلاح برابر با همان "مدیر اسکرام اسکرام" ها "یا" رئیس" یا "مدیر اسکرام اصلی" و غیره می باشد. او هیچ تیم را به صورت تک هدایت نمی

کند. بلکه او مسئول مسائل فراتیمی می باشد مثلاً چه کسی باید مدیر اسکرام تیم شود و یا چه کسی باید در چه تیمی قرار بگیرد و یا غیره.

ما با تلاش های فراوان توانستیم یک نام خوب برای این نقش پیدا کنیم که بالاخره با عنوان "رهبر تیم" موافقت کردیم.

این راه حل برای ما کارآمد بوده است و این را می توانم برای شما هم پیشنهاد دهم.

چگونه نفرات را بین تیم ها تقسیم می کنیم؟

زمانیکه چندین تیم در محیط اسکرام وجود دارد دو روش کلی برای تقسیم نفرات بین تیم ها وجود دارد:

- برای اختصاص نفرات به تیم کسی را تعیین کنید مثلاً همان "رهبر تیم".
- به تیم ها اجازه بدهید تا خودشان یک طوی این کار را انجام بدهند.

ما از هر 3 این روش ها استفاده کرده ایم. سه؟ بلی. روش 1 و روش 2 و ترکیب روش های 1 و 2. به این نتیجه رسیدیم که ترکیب این دو روش بسیار عالی عمل می کند.

قبل از جلسه برنامه ریزی اسپرینت، رهبر تیم، مالک محصول و تمام مدیران اسکرام را برای جلسه اختصاص دادن افراد به تیم ها دعوت می کند. ما در این جلسه بر روی آخرین اسپرینت متمرکز می شویم. به دنبال تیم ها مجازی می گردیم در صورت نیاز تیم ها را تقسیم و یا تجمیع می کنیم. ما بر روی تقسیم بندی توافق می کنیم و آن توافق را با عنوان پیشنهاد اختصاص تیم ها یادداشت می کنیم و این را با خودمان به جلسه برنامه ریزی اسپرینت می بریم.

جلسه برنامه ریزی اسپرینت همانند قبل اجرا می شود. ولی در آخر رهبر تیم مثلاً می گوید: "سلام به همه، ما برای اسپرینت بعدی پیشنهاد می کنیم که تیم ها بدین حالت در بیایند."

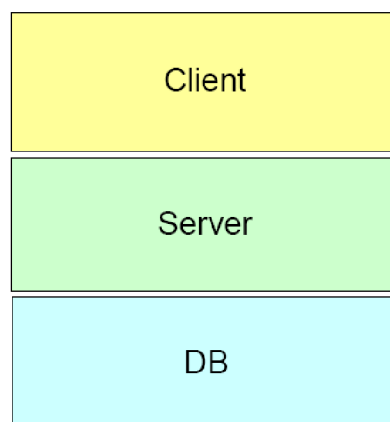
تقسیم بندی اولیه تیم ها		
Team 1 - tom - jerry - donald - mickey	Team 2 - goofy - daffy - humpty - dumpty	Team 3 - minnie - scrooge - winnie - roo

"همانطور که ملاحظه می کنید تعداد تیم ها از 4 تیم به 3 تیم کاهش یافته است. لطفا بایستید و در کنار گروه های خود قرار بگیرید."

این تقسیم بندی یک تقسیم بندی اولیه می باشد و در طی جلسه برنامه ریزی اسپرینت قابل تغییر خواهد بود. شما می توانید همکاری نفرات تیم های جدید را با هم مشاهده کنید و یا بر اساس تعاملات آنها تیم های جدید به وجود آورید یا در طی برنامه ریزی اسپرینت بر اساس الویت های مالک محصول نفرات تیم ها را تعویض کنید.

تیم های تخصصی آری یا نه ؟

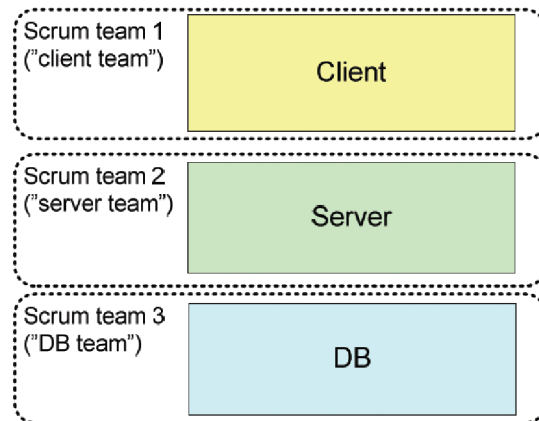
فرض کنید فناوری شما از 3 جزء زیر تشکیل شده باشد:



و فرض کنید که 15 نفر نیروی کار دارید که برای این محصول اختصاص داده شده اند. به احتمال زیاد که همه 15 نفر طی یک تیم کار نخواهند کرد پس چه تیم هایی به وجود خواهید آورد؟

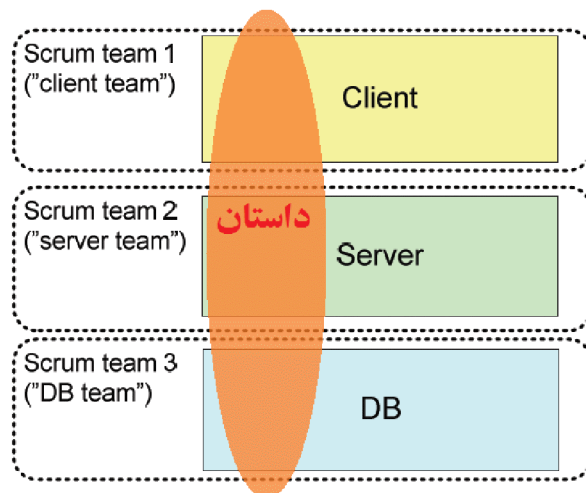
روش 1: تیم های تخصصی اجزا

یک روش این است که تیم های تخصصی ایجاد کنید که هرکدام متخصص و متعهد جزئی از محصول باشند مثلاً یک "تیم Client" یک "تیم Server" و یک "تیم دیتابیس".



ما در ابتدا با این روش شروع کردیم ولی زمانیکه داستانی مابین تمام این اجزا و یا در کل این اجزا بود این روش خیلی خوب کار نمی کرد.

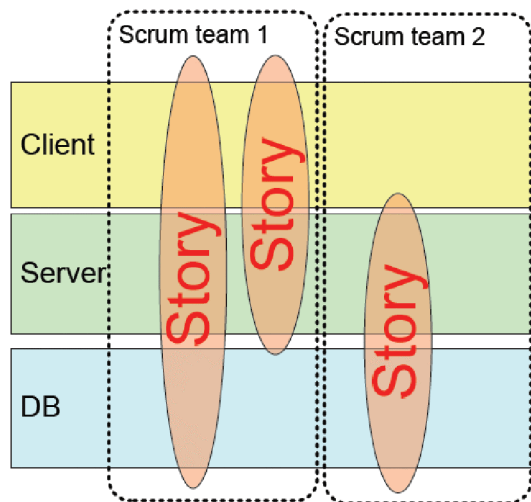
برای مثال فرض کنید یک داستان با عنوان "ایجاد تابلوی اعلانات برای اینکه کاربران بتوانند به یکدیگر پیام بفرستند" داریم. این ویژگی تابلوی اعلانات شامل آپدیت و اضافه کردن در دیتابیس و یک رابط کاربری و تغییر بر روی لایه سرور و اضافه کردن جدول جدید در دیتابیس خواهد بود.



و این بدین معنی می باشد که هر 3 تیم کلاینت - سرور و دیتابیس باید برای انجام این داستان همکاری و تعامل داشته باشند که خوب به نظر نمی رسد.

روش 2: تیم های فرا اجزا

روش دوم این است که تیم هایی ایجاد کنیم به جای اینکه تخصصی بر روی یک اجزا کار کنند بتوانند در همه اجزا کار کنند.



اگر بسیاری از داستان های شما بدین گونه در بین اجزای مختلف باشند این روش می تواند موثر عمل کند. هر تیم می تواند یک داستان را به طور کامل در بخش های مختلف اعم از کلاینت - سرور و یا دیتابیس تکمیل نماید. به این صورت تیم ها خواهند توانست که بدون وابستگی به یکدیگر کار کنند که این بسیار خوب است.

اولین کار در زمان استفاده اولیه از اسکرام تبدیل تیم های اجزا گرا به تیم فرا اجزا بود. در تیم های اجزا گرا معمولاً با چنین صحبت هایی سروکار داشتیم: "تا زمانیکه تیم سرور کار خود را تمام نکند ما نمی توانیم کار خود را بر روی این داستان انجام دهیم." این علاوه بر وابستگی باعث ایجاد تاخیر هم می شود.

البته برخی اوقات برای نیازهای بدون وابستگی گاهی از تیم تخصصی اجزا گرا نیز استفاده می کنیم.

دوباره چینی تیم ها در بین اسپرینت ها آری یا نه ؟

معمولا هر اسپرینت با اسپرینت های دیگر بدلیل وجود داستان های مختلف یا اهداف مختلف متفاوت می باشد به همین دلیل می توان از راه اندازی تیم های متفاوت برای هر اسپرینت استفاده کرد.

در واقع , در اکثر اسپرینت ها می گوئیم که این اسپرینت یک اسپرینت نرمال نیست بدلیل بعد از مدتی ما به همین اسپرینت غیر نرمال مارک نرمال بودن را خواهیم زد. در واقع هیچ اسپرینت غیر نرمالی وجود ندارد. همانطور که در واقع هیچ انسان و یا خانواده غیر نرمالی وجود ندارد.

شاید نسبت به داستان های داخل اسپرینت بتوان تیم های جدیدی را تشکیل داد. مثلا اگر داستان ها بیشتر مربوط به کلاینت باشند می توان یک تیم کلاینت تخصصی ایجاد کرد و به طور تخصصی این تیم مسئول داستان های کلاینت خواهد بود.

یکی از نمودهای کلیدی اسکرام ژل شدن تیم می باشد. اگر اعضای یک تیم در طی چندین اسپرینت متوالی با هم کار کنند آنها خیلی به هم نزدیک خواهند شد. آنها یاد خواهند گرفت که چگونه به چرخه گروهی دست یابند و چگونه به صورت تیمی بهروری خود را ارتقاء دهند. اما این نیاز به باهم بودن اعضای تیم در طی زمان و اسپرینت های متوالی دارد. اگر قرار باشد دائما این تیم ها تغییر کنند هیچ گاه نمی توان به تیم های ژل شده دست یافت.

نفرات پاره وقت

من فقط می توانم در این مورد نظرات کتاب های اسکرام را قبول کنم که نفرات پاره وقت ایده خوبی نیستند.

فرض کنید که شما در حال جذب "جو" به عنوان یک نیروی پاره وقت برای تیم اسکرام خود می باشید. آیا واقعا به جو نیاز دارید؟ آیا مطمئن هستید که نمی توانید جو را به صورت تمام وقت در اختیار داشته باشید؟ مسئولیت های دیگر او چه هستند؟ آیا کس دیگری می تواند مسئولیت های دیگر جو را متعهد شود تا جو با واگذاری این مسئولیت به دیگری بتواند کمی بیشتر فعال باشد؟ آیا جو خواهد توانست از اسپرینت بعدی به صورت تمام وقت حاضر شود؟

برای مسائلی راه دیگری وجود ندارد. شما ضرورتاً به جو نیاز دارید زیرا جو تنها DBA موجود در ساختمان می باشد. اما تیم های دیگر هم به او نیاز دارند به همین دلیل او نمی تواند وقت خود را به صورت تمام وقت در اختیار تیم شما قرار دهد و شرکت توانایی استخدام DBA جدید را ندارد. خوب. این دلیل خوبی است که بتوانید از یک نیروی کار پاره وقت استفاده کنید ولی دقت کنید که ضرورت وجود نیروی پاره وقت را همیشه ارزیابی کنید.

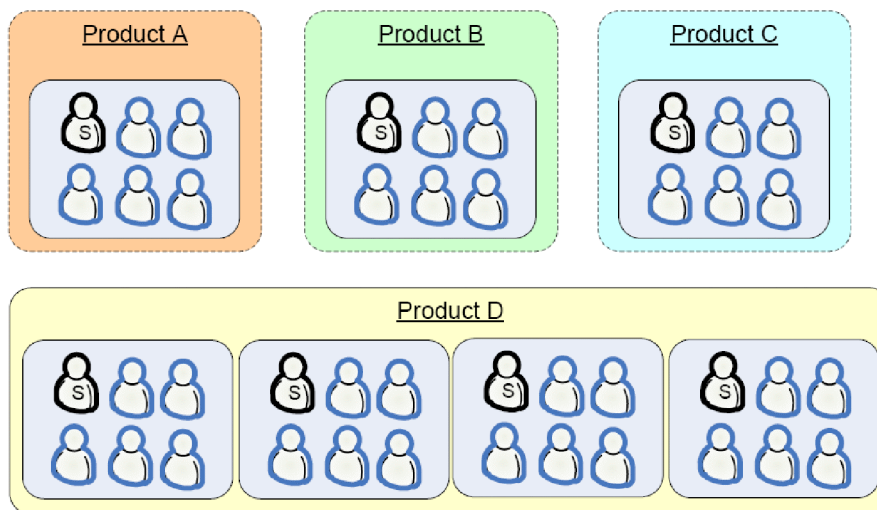
به شخصه من ترجیح می دهم به جای 8 نفر نیروی پاره وقت 3 نفر نیروی تمام وقت در اختیار داشته باشم.

حتی اگر نفراتی داشته باشید مانند DBA مثال زده شده که باید وقت خود را بین تیم های مختلف تقسیم کنند، می توانید برای او یک تیم خانه انتخاب کنید. یعنی هر کدام از تیم ها که به او بیشتر نیاز دارند، همان تیم خانه او شود و اولویت اصلی برای نفر آن تیم خواهد شد و زمانیکه تیم خانه به او نیاز نداشت می تواند به تیم های دیگر سر بزند.

چگونه اسکرام اسکرام ها را انجام می دهیم

اسکرام اسکرام ها به صورت طبیعی یک جلسه می باشد که تمام مدیران اسکرام برای صحبت جمع می شوند.

فرض کنید ما 4 محصول به طور همزمان داریم. همه محصول ها دارای یک تیم با یک مدیر اسکرام می باشند ولی محصول آخر دارای چندین تیم با چندین مدیر اسکرام می باشد. مثلاً مانند این عکس:



این عکس به این معنی می باشد که ما دو سطح اسکرام اسکرام ها خواهیم داشت. یکی در سطح محصول D که تمام مدیران اسکرام محصول D آن را برگزار خواهند کرد. و یک جلسه اسکرام اسکرام ها در سطح شرکت که با حضور تمام مدیران اسکرام محصول ها انجام خواهد شد.

اسکرام اسکرام های سطح محصول

این جلسه بسیار مهم می باشد. این جلسه معمولاً هر هفته یکبار انجام می شود. ما در مورد مسائل یکپارچگی، مسائل بالانس تیم، آماده شدن برای جلسه برنامه ریزی اسپرینت بعدی و غیره بحث می کنیم. ما معمولاً برای این جلسه 30 دقیقه در نظر می گیریم که معمولاً کمی بیشتر می شود.

دستور جلسه اسکرام اسکرام های ما بدین صورت می باشد:

1. در یک میزگرد، هر کس توضیح می دهد که اعضای تیم او در هفته قبل چه چیزهایی را تکمیل کردند، این هفته چه چیزهایی را قصد دارند تکمیل نمایند و مشکلات و مسائل پیش بینی نشده آنها چه بوده است.
2. بحث در مورد مسائل فراتیمی که باید حل شوند مثلاً مسائل یکپارچگی.

صورت جلسه این جلسه خیلی برای من مهم نیست و مورد مهم برای من این است که این جلسات اسکرام اسکرام ها تشکیل شوند.

اسکرام اسکرام های سطح شرکت

ما این جلسه را تپش یا پالس می نامیم. این جلسه را در قالب های متفاوت اجرا کرده ایم و در نتیجه به این مسئله رسیده ایم که این جلسه باید به صورت هفتگی با حضور همه (همه کسانی که در توسعه حضور دارند) به مدت 15 دقیقه تشکیل بشود.

15 دقیقه؟ همه؟ آیا امکان دارد!؟

بلی و بسیار کارا می باشد و مسئله حیاتی کوتاه بودن جلسه می باشد.

قالب جلسه این است که :

1. اعلان خبرها و بروز رسانی های مهم در توسعه. برای مثال رویدادهای پیش رو.
2. گردش به نوبت. هر یک از نمایندگان گروه ها گزارش خود را در مورد اینکه هفته پیش چکار انجام داده اند و این هفته چه می خواهند انجام بدهند و مشکلات پیش رو را ارائه می دهند. نفراتی دیگری هم در جلسه گزارششان را ارائه می دهند مانند QA.
3. نفرات دیگر حاضر در جلسه آزاد به اضافه کردن توضیحات و پرسیدن سوال می باشند.

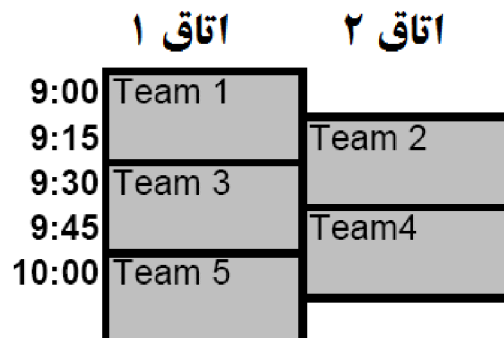
این یک تشکل برای اطلاعات مختصر و مفید می باشد و نه بحث و بازخورد. معمولا 15 دقیقه برای این جلسه کافی می باشد و البته معمولا بیشتر از 30 دقیقه به طول نمی انجامد. ولی اگر بحث جالب بود و بیشتر از این طول کشید من جلسه را قطع می کنم و از علاقمندان دعوت می کنم که برای ادامه جلسه فردا تشریف بیاورند.

چرا ما از همه در جلسه تپش استفاده می کنیم؟ به این خاطر که جلسه اسکرام اسکرام ها بیشتر در مورد ارائه گزارشها می باشد. به عبارتی تیم های دیگر بدین صورت مطلع خواهند شد که دیگران در حال انجام چه کاری هستند و این باعث به وجود آمدن شفافیت در سطح کل شرکت خواهد شد.

جلسات روزانه اسکرام در زمان ها متفاوت

اگر شما چندین تیم اسکرام برای انجام یک محصول داشته باشید و همه آنها جلسه روزانه اسکرام را در یک زمان انجام بدهند با یک مشکل برخورد خواهید کرد: فقط مالک محصول در یک جلسه می تواند حاضر شود.

برای رفع این مشکل ما از تیم ها می خواهیم که جلسات روزانه اسکرام را در یک زمان انجام ندهند.



زمانبندی ذکر شده برای حالتی است که ما جلسات روزانه اسکرام را در اتاق دیگری جز اتاق تیم برگزار می کردیم. زمان هر جلسه 15 دقیقه بود ولی هر تیم 30 دقیقه در اتاق حضور داشت.

این روش شدیداً سودمند می باشد زیرا :

1. افراد و مالک محصول دوست دارند که جلسات روزانه اسکرام دیگران را ببینند.
2. تیم ها همدیگر را می بینند و بیشتر تعامل برقرار می کنند. البته بیشتر دو تیم .

بدی این روش این می باشد که آزادی انتخاب زمان جلسه برای تیم از بین می رود و آنها همیشه باید در یک زمان مقرر این جلسه را برگزار کنند.

تیم های مبارزه با آتش

ما مواردی را دیده ایم که مثلاً در یک محصول بزرگ امکان پیاده سازی اسکرام وجود نداشت زیرا تیم بیشتر زمان خود را صرف مبارزات با آتش مثلاً رفع باگ های وحشتناک در زمان Release سیستم می کرد. این واقعا یک چرخه نادرستی است ، آنها آنقدر زمان خود را صرف این مبارزه می کردند که زمان این را نداشتند که از به وجود آمدن این مبارزه با آتش جلوگیری کنند مثلاً بهبود طراحی ، تست های اتوماتیک ، ایجاد ابزار مانیتورینگ ، ابزار هشدار دهنده و ...

ما این مشکل را با ایجاد تیم های اسکرام مبارزه با آتش حل کردیم. به این منظور تیم هایی را فقط برای مبارزه منصوب کردیم.

وظیفه تیم های اسکرام این است که سیستم را تثبیت نمایند و به طور موثر از ایجاد شدن باگ ها (آتش ها) بپرهیزند.

تیم های مبارزه با آتش (ما به این ها تیم پشتیبان می گوئیم) دو وظیفه اصلی دارند :

1. مباره با آتش (رفع باگ های خطرناک)
2. پوشش و محافظت از تیم های اسکرام در مقابل هر نوع اختلالی از هر جایی.

جایگاه تیم مبارزه با آتش در کنار در می باشد و تیم اسکرام در انتهای اتاق مستقر می شود. حتی تیم های مبارزه با آتش می توانند از تیم های اسکرام در مقابل اختلالات فیزیکی مانند مشتری خشمگین و یا مدیر فروش عصبانی محافظت نمایند.

توسعه دهندگان ارشد در هر دو تیم قرار می گیرند. به همین دلیل یک تیم به تیم دیگری زیاد وابسته نمی شود.

تیم های مبارزه با آتش کاملاً مسئول رفع باگ ها و هر نوع اختلال در عملکرد تیم های اسکرام می باشند و این باعث خواهد شد که تیم های اسکرام بتوانند فقط در فکر انجام کار خودشان باشند. البته به این صورت نخواهد بود که تیم از هر نوع اختلالی بر حذر باشد ولی این اختلالات به حداقل خود خواهند رسید.

بعد از چند ماه که سیستم به یک پایداری خوبی رسید، می توانیم تیم های مبارزه با آتش را منحل و تیم های جدید اسکرام را به وجود آوریم.

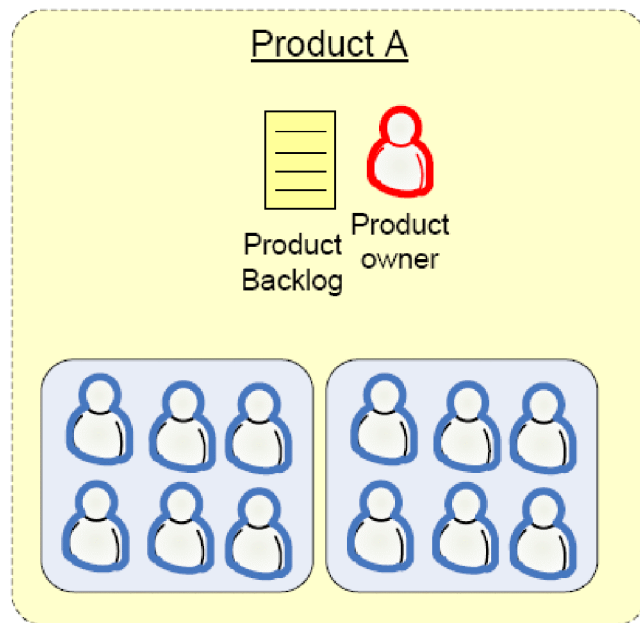
جداسازی بک لاگ محصول – آری یا نه ؟

فرض کنید یک محصول همراه با دو تیم اسکرام دارید. چند بک لاگ محصول باید وجود داشته باشد؟ چند مالک محصول باید وجود داشته باشد؟ ما ارزیابی کردیم که برای حل این 3 مدل وجود دارد. انتخاب هر مدل باعث تغییراتی در انجام برنامه ریزی اسپرینت خواهد شد:

روش 1: یک مالک محصول و یک بک لاگ محصول

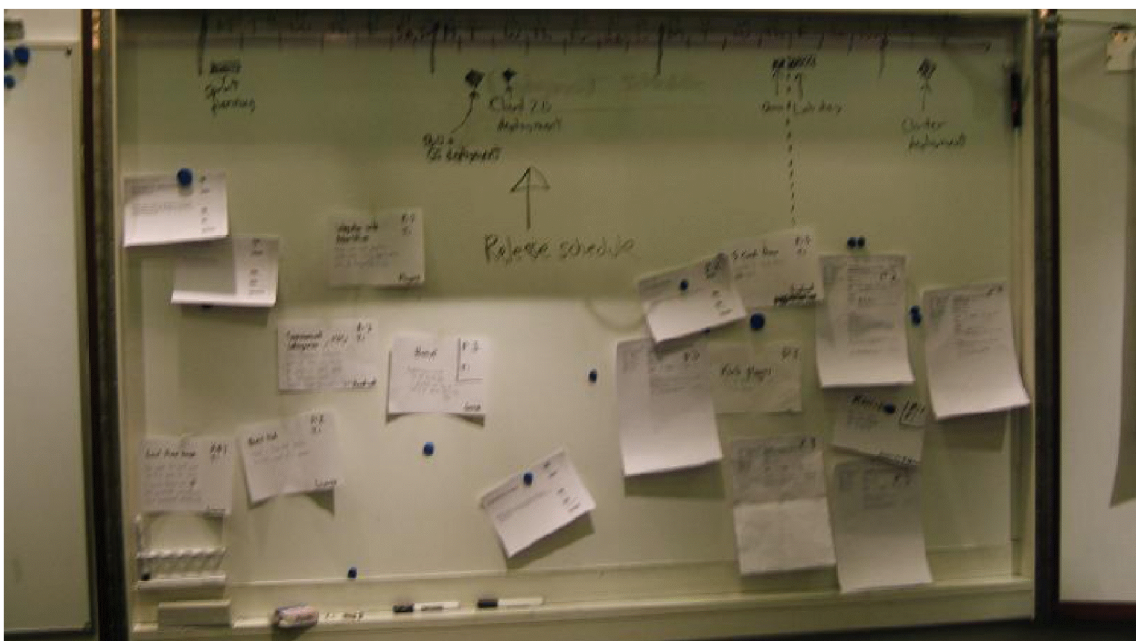
این روش مورد علاقه ما می باشد.

در این روش دقت و تمرکز بدلیل وجود یک نفر و یک بک لاگ بالا خواهد بود. مالک محصول به خوبی خواهد توانست که بر روی خواسته های خود تمرکز کند و آن ها را از تیم طلب نماید و تیم بدلیل وجود یک بک لاگ خواهد توانست الویت های مالک محصول را به خوبی درک و پیاده سازی نماید.



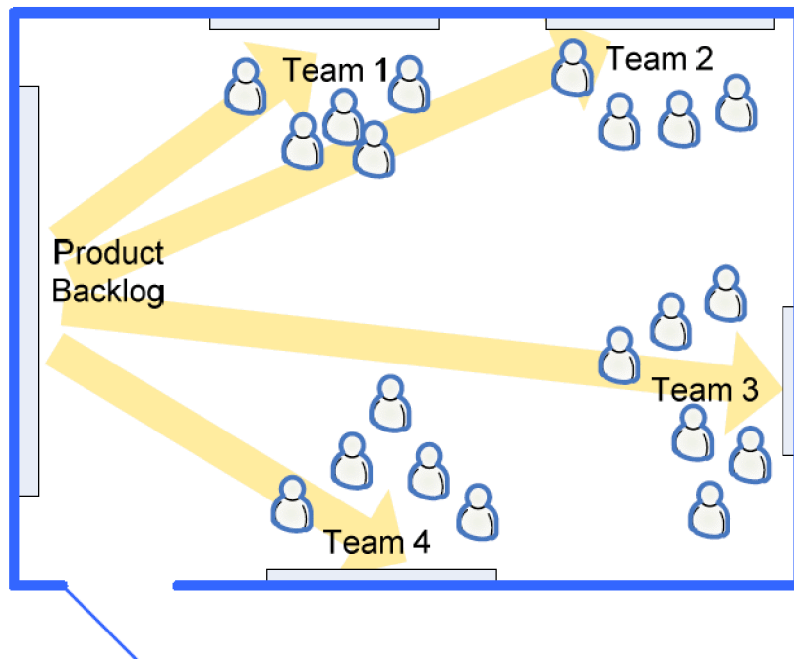
اما جلسه برنامه ریزی اسپرینت در این حالت چگونه انجام می شود؟
جلسه برنامه ریزی در یک جا مانند اتاق کنفرانس برگزار می شود.

قبل از شروع جلسه مالک محصول یک دیوار بک لاگ محصول انتخاب می نماید و داستان ها (کارت های ایندکس) را به ترتیب اهمیت و اولویت انتخاب و بر روی دیوار قرار می دهد. او این کار را تا زمانی انجام می دهد که دیوار پر شود معمولاً باید به اندازه یک اسپرینت داستان بر روی دیوار قرار بگیرد.



هر تیم اسکرام در یک دیوار جالی خالی پیدا می کنند و نام تیم خود را در روی آن جای خالی قرار می دهد. این دیوار تیم آنها می باشد. هر تیم داستان هایی را از دیوار بک لاگ جمع آوری و در دیوار خود قرار می دهد البته با رعایت الویت داستان ها.

این کار کاملا در عکس زیر مشخص می باشد. پیکان ها از چرخه انتخاب و قرار گیری داستان ها از دیوار بک لاگ محصول در دیوار تیم حکایت می کنند.



در ادامه جلسه ، مالک محصول و تیم ها بر روی کارت ایندکس چانه زنی می کنند و کارت ها بین تیم ها جابجا می شود، یا اولویت داستان ها تغییر پیدا می کنند ، آن ها به داستان کوچک تر شکسته می شوند و غیره. بعد از یک ساعت یا کمی بیشتر هر تیم یک نسخه کاندید از بک لاگ اسپرینت بر روی دیوار خود خواهد داشت. بعد از این تیم ها به صورت جداگانه و غیر وابسته مشغول برآورد و شکستن داستان به وظایف خواهند شد.

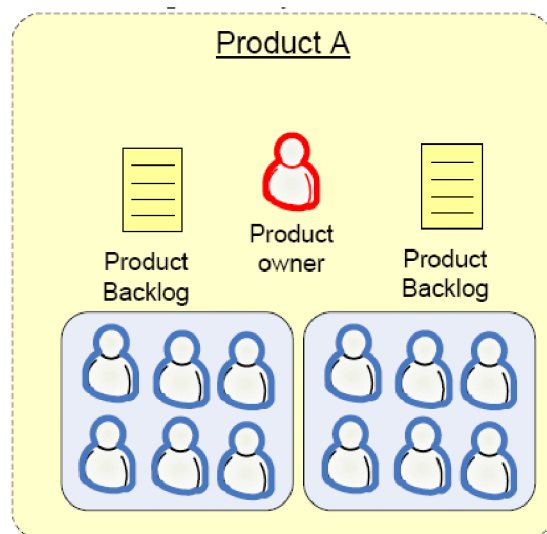


با وجود اینکه این جلسه کمی شلوغ و آشفته می باشد ولی باز یک جلسه تاثیر گذار همراه با شادی خواهد بود. زمانی که وقت جلسه تمام شد معمولا تیم ها به اندازه کافی برای شروع اسپرینت اطلاعات خواهند داشت.

روش 2: یک مالک محصول و چندین بک لاگ

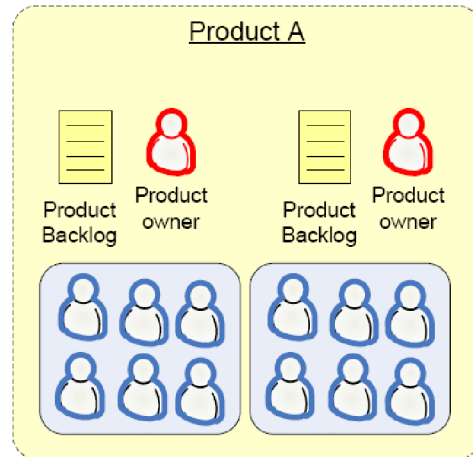
در این روش مالک محصول باید چندین بک لاگ محصول را نگه داری کند. این روش را ما در عمل آزمایش نکردیم ولی روش شبیه به این را انجام دادیم که منجر به شکست شده بود.

ضعف اصلی این روش این است که عمل اختصاص وظایف به تیم ها توسط مالک محصول انجام خواهد شد که این را تیم بهتر می تواند انجام بدهد.



روش 3: چندین مالک محصول و به ازای هر مالک یک بک لاگ

این روش از روش دوم بهتر می باشد.



اما این روش را ما انجام نداده ایم و احتمالاً هم هیچ وقت انجام نخواهیم داد .

اگر دو بک لاگ محصول شامل یک کد پایه باشند ، این باعث به وجود آمدن تداخل جدی بین دو مالک محصول خواهد شد.

اگر دو بک لاگ محصول شامل کدهای جدا از هم باشند این دقیقاً مانند دو تکه کردن محصول به دو محصول کوچکتر می باشد که هر تیم مسئول یک محصول می شود و در این حالت تیم ها هم می توانند جدا از هم به راحتی کار کنند.

انشعاب کد

با وجود تیم های متفاوتی که بر روی یک کد کار می کنند مجبوریم با انشعاب کد در سیستم SCM (مدیریت پیگیربندی نرم افزار³¹) روبرو شویم. کتاب ها و مقالات زیادی در مورد نحوه مدیریت کدهای مشترک بین افراد مختلف وجود دارد و من نمی خواهم به مسائل جزئی وارد شوم و فقط خلاصه ای از تجربیات خودمان در این مورد را ارائه می دهم :

³¹ Software Configuration management

- در مورد نگه داری خط اصلی یا Trunk در وضعیت سازگار محکم عمل کنید. به این معنی که هر چیز باید حداقل کامپایل شود و تست ها باید پاس شوند. باید در هر زمان بتوان یک نسخه کارآ از سیستم خروجی گرفت. سیستم های ساخت متوالی می توانند زنگ خطر این مورد را برای ما به صدا در آورند.
- برچسب زدن هر Release . در زمان Release کردن برای تست پذیرش و یا برای مشتری مطمئن شوید که نسخه مربوطه در Trunk برچسب زده شود تا بعدا دقیقا قابل فهم باشد که در این ارائه چه چیزی Release شده است. در این حالت می توانید در هر نقطه به راحتی به عقب بازگردید.
- انشعابات جدید را فقط زمانی ایجاد کنید که واقعا نیاز باشند. این به این خاطر می باشد که ایجاد هر انشعاب هزینه مدیریت آن را به همراه خواهد داشت.
- از انشعاب اساسا برای جداسازی چرخه های حیات متفاوت استفاده کنید. مثلا باگ زدائی های کوچک با باگ زدائی های بزرگ.
- همسان سازی کنید. اگر به صورت انشعاب کار می کنید هر زمان که چیزی ساخته شد خط اصلی را همسان سازی نمایید. معمولا همسان سازی نسخه اصلی را در برنامه خود قرار دهید.

بازبینی چندین تیم

ما چگونه زمانیکه چندین تیم بر روی یک محصول کار می کنند جلسه بازبینی اسپرینت را انجام می دهیم؟

بعد از دمو ، تیم ها بازبینی اسپرینت را طبق توضیحات بخش "چگونه اسپرینت را بازبینی می کنیم" انجام خواهند داد.

بعد از اینکه هر تیم جلسه بازبینی خود را انجام داد یک جلسه با حضور همه تیم ها با عنوان جمع بندی بازبینی برگزار می شود. در طی این جلسه ، ما به سخنگویان هر تیم حدودا 5 دقیقه وقت می دهیم تا بتوانند مسائل کلیدی که می توان بهبود داد را بیان کنند.

این جلسه در حدود 20 دقیقه به طول خواهد کشید. اما بعد از یک استراحت کوتاه ، تیم ها بلافاصله جلسه برنامه ریزی اسپرینت بعدی را شروع خواهند کرد.

در حالت چندین تیم ما فرصت و فاصله بین اسپرینت آنچنان نداریم. مهمترین ایراد این روش هم همین است که گپی بین اسپرینت ها وجود ندارد.

و البته برای محصول های یک تیمی نیازی به برگزاری جلسه جمع بندی بازینی اسپرینت ها نمی باشد.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

16

چگونه تیم های توزیع شده جغرافیایی را اداره می کنیم

زمانی که اعضای تیم از نقاط جغرافیایی متفاوت باشند چه اتفاقی خواهد افتاد؟ بیشتر سحر و جادوی اسکرام و XP مربوطه به نزدیکی و همکاری اعضای تیم به صورت ارتباط چهره به چهره یا برنامه نویسی جفتی می باشد.

ما تیم هایی داریم که در نقاط مختلف جهان قرار دارند و یا حتی نفراتی داریم که در خانه خود مشغول به کار می باشند.

استراتژی ما برای اداره کردن این تیم ها بسیار ساده می باشد. ما از هر روشی که بتوان پهنای باند ارتباطی تیم های غیر هم مکان را به حداکثر رساند استفاده می کنیم. البته منظور من از پهنای باند فقط مگابیت/ثانیه نبود. منظور من از پهنای باند ارتباطی می تواند چنین چیزهایی باشد:

- قابلیت برنامه نویسی جفتی باهم.
- قابلیت ملاقات همدیگر به صورت چهره به چهره در جلسات روزانه اسکرام.
- قابلیت داشتن ملاقات های چهره به چهره در هر زمان.
- قابلیت داشتن ملاقات فیزیکی و اجتماعی.
- قابلیت برگزاری جلسات فوری با حضور تمام اعضای تیم.
- قابلیت دیدن یک بک لاگ اسپرینت , نمودار Burndown و

بعضی از راحل هایی که ما انجام داده ایم :

- تهیه وب کم و هدست برای هر ایستگاه کاری.
- اتاق های کنفرانس همراه با وب کم و میکروفن , ... با قابلیت دسترسی از راه دور.

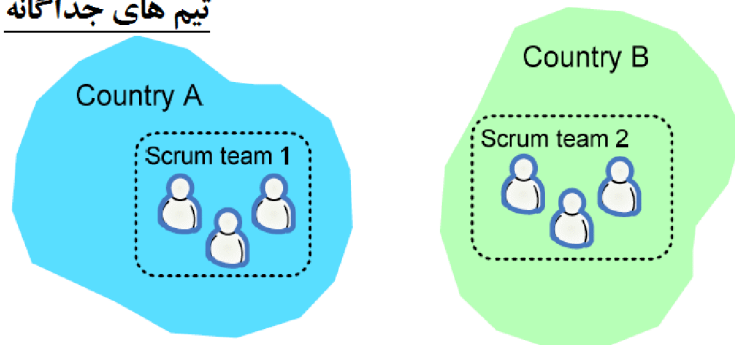
- "پنجره های راه دور". صفحه های بزرگی که می توان محل کار تیم های دیگر را در آن مشاهده کرد. مانند یک پنجره مجازی که در داخل یک دپارتمان قرار دارد. مثلا می توان در این صفحه با اعضای آن تیم صحبت کرد.
 - برنامه های مبادله که اعضای تیم ها بتوانند به محل دیگر تیم ها سفر بکنند و از نزدیک نفرات دیگر را در محلشان ملاقات کنند.
- درست است که با استفاده از این امکانات نیز سرعت تیم کم خواهد بود ولی با وجود امکانات ذکر شده تیم ها خواهند توانست که در نقاط مختلف جهان با هم کار کنند.

البته تکنیک های جدید را باید طی استفاده و تجربه بدست آورید . کنترل و رسیدگی => سازگار کردن => کنترل و رسیدگی => سازگار کردن => کنترل و رسیدگی => سازگار کردن

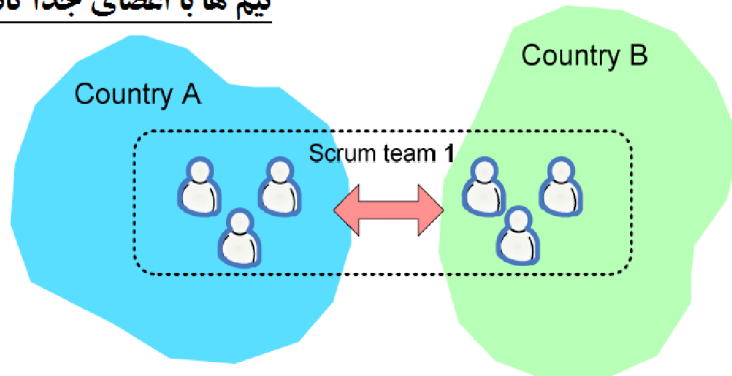
دور از ساحل

ما چندین نوع تیم دور از ساحل (منظور پراکنده در نقاط مختلف جغرافیایی) را تجربه کرده ایم. به طور کلی دو استراتژی برای این کار وجود دارد: تیم های جداگانه و تیم ها با اعضای جدا از هم.

تیم های جداگانه



تیم ها با اعضای جداگانه



استراتژی اول به نظر ما بهتر است از دومی می باشد و به همین دلیل هیچ وقت از روش دوم استفاده نکرده ایم. در زیر تعدادی دلیل برای استفاده نکردن از روش دوم ذکر شده است:

1. ما می خواهیم که اعضای تیم ها همدیگر را به خوبی بشناسند.
2. ما می خواهیم علاوه بر مسئله فرای تیمی ، حداقل بین اعضای تیم ارتباط موثر و عالی برقرار باشد.
3. در زمان شروع ، تیم کنار از ساحل بسیار کوچک تر از آن خواهد بود که بتواند یک تیم موثر اسکرام تشکیل شود.
4. ما می خواهیم دائما بین اعضای تیم دانش و اطلاعات رد و بدل شود.

پس به همین خاطر تیم های جدا بهتر از تیم هایی خواهند بود که اعضای آن جداگانه کار می کنند.

اعضای تیم در خانه اشان کار می کنند

کار کردن اعضای تیم در خانه بعضی از اوقات می تواند ایده خوبی باشد. شاید در یک روز بتوانید از یک هفته کار در محل شرکت بیشتر برنامه نویسی کنید. البته اگر بچه نداشته باشید.

اما هنوز هم یکی از اصلی ترین موارد اسکرام در کنار هم بودن فیزیکی اعضای تیم می باشد. برای این چکاری می کنیم؟

معمولا ما این را به تیم واگذار می کنیم تا خودشان تصمیم بگیرند که چه وقت هایی می توانند در خانه کار کنند. البته ما بیشتر تیم ها را تشویق می کنیم که اکثرا به صورت حضور فیزیکی در یک محل کار کنند.

زمانیکه اعضای تیم در خانه هایشان کار می کنند ، توسط تماس صوتی اسکایپ در جلسه روزانه اسکرام شرکت می کنند. آن ها در طول روز همیشه توسط سیستم های پیام دهی فوری آنلاین می باشند. البته این به خوبی هم نشینی در یک اتاق نمی باشد ولی باز بد نیست.

ما یکبار سعی کردیم یک روز را به عنوان روز توجه تعیین کنیم تا هر کس که خواست بتواند این روز را در خانه کار کند. روز تعیین کرده ما چهارشنبه بود. البته همین چهارشنبه هم باید با تیم هماهنگ می شد تا نفر بتواند در خانه کار کند. نتیجه این تصمیم هم خوب بود و اکثر اعضای تیم در روز چهارشنبه در خانه می ماندند و البته کارهای بیشتری را نسبت به بقیه روزها انجام می دادند. ولی فقط این یک روز بود و نه بیشتر.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

17

چک لیست مدیر اسکرام

در فصل آخر من چک لیست مدیر اسکرام خودمان را نشان شما خواهم داد که شامل لیست کارهای مدیریتی معمول مدیر اسکرام می باشد. مواردی که بسیار راحت فراموش می شوند. مثلاً مسئله معلوم و مهم "حذف موانع از پیش روی تیم" را به راحتی فراموش می کنیم.

شروع اسپرینت

- بعد از جلسه برنامه ریزی اسپرینت صفحه اطلاعات اسپرینت را ایجاد کنید.
 - لینک صفحه اطلاعات خود را در داشبورد ویکی اضافه کنید.
 - از صفحه مربوطه پرینت بگیرید و در جلوی اتاق تیم نصب کنید.
- یک ایمیل برای همه ارسال کنید و اطلاع دهید که اسپرینت جدید شروع شده است که این ایمیل شامل شرح هدف اسپرینت و لینک صفحه بر روی ویکی خواهد بود.
- مستند آمار اسپرینت را آپدیت نمایید. سرعت برآورد شده، اندازه تیم، طول اسپرینت خود را در داخل آن اضافه کنید.

طی اسپرینت

- مطمئن شوید که جلسه روزانه اسکرام در زمان مشخص شروع و پایان پذیرد.
- مطمئن شوید که داستان ها برای نگه داری اسپرینت در زمانبندی، اضافه یا حذف شوند.
 - مطمئن شوید که مالک محصول از این تغییرات مطلع شود.
- مطمئن شوید که بک لاگ اسپرینت و نمودار Burndown توسط تیم به روز نگه داشته شوند.
- مطمئن شوید که مشکلات و موانع یا حل شوند و یا به مالک محصول یا مدیر بخش توسعه گزارش داده شوند.

انتهای اسپرینت

- دموی اسپرینت را به صورت باز و با حضور همه ارائه دهید.
- همه باید در مورد زمان و مکان ارائه دموی یک یا دو روز قبل مطلع شوند.
- جلسه بازیابی اسپرینت را با حضور تیم و مالک محصول برگزار کنید. می توانید برای این جلسه رئیس بخش توسعه را نیز دعوت کنید.
- مستند آمار اسپرینت را با اضافه کردن سرعت واقعی تیم و نقاط کلیدی بازیابی اسپرینت به روز رسانی کنید.

نسخه الکترونیکی رایگان - برای تهیه نسخه چاپی و یا حمایت از اثر به لینک زیر مراجعه کنید:

<http://www.irAgile.com>

18

سخن آخر

اصلاً فکر نمی کردم که اینقدر به طول بینجامد.

امیدوارم این کتاب بتواند ایده های خوبی را به شما بدهد تا بتوانید اسکرام را به خوبی انجام دهید.

به هر حال هیچ وقت نمی توان گفت که بهترین روش برای انجام اسکرام چه چیزی می باشد، بدلیل اینکه وضعیت در هر جا با جاهای دیگر متفاوت می باشد. ولی من خوشحال خواهم شد که بازخورد های شما را دریافت نمایم. روش هایی را که برای انجام اسکرام انجام می دهید و با روش های ذکر شده با این کتاب متفاوت می باشند را برای من بفرستید.

می توانید با این ایمیل تماس بگیرید : henrik.kniberg@crisp.se

هم چنین نیم نگاهی به یاهو گروه scrumdevelopment@yahoogroups.com دارم.

اگر این کتاب را دوست داشتید پیشنهاد می کنم بعضی اوقات وبلاگ من را نیز مطالعه بفرمایید. در این وبلاگ پست هایی در مورد Java و Agile ارسال می کنم.

<http://blog.crisp.se/henrikkniberg>

هنریک کنیبرگ



معرفی IR Agile

تشکل IR Agile اولین و تنهاترین گروه چابک سازی طی ارائه آموزش و مشاوره در کل ایران با حمایت Scrum Alliance و Agile Alliance می باشد. حیطه کاری این گروه در عرصه چابک سازی صنعت توسعه نرم افزار می باشد که خدمات آن قابل ارائه برای سازمان ها، شرکت ها و متخصصین صنعت توسعه نرم افزار می باشد. این گروه شامل متخصصین و مربیان Agile از سراسر ایران که آموزش های لازم را در محافل بین المللی توسط اساتید جهانی این حوزه گذرانده اند می باشد.

این گروه به طور رسمی فعالیت خود را در اواخر سال 88 شروع کرده است و طی این زمان اقدام به گردآوری و آموزش مربیان از اقصی نقاط کشور کرده است. از اولین اقدامات رسمی اعضای این گروه می توان به تولید محتوا و آموزش و از همه مهمتر معرفی تفکر Agile در ایران اشاره کرد. یکی از افتخارات اصلی اعضای این گروه معرفی و آموزش Agile در ایران برای اولین بار می باشد که در طی این پروسه چندین مقاله مهم به رشته تحریر درآمد و در اختیار متخصصین توسعه نرم افزار قرار گرفت.

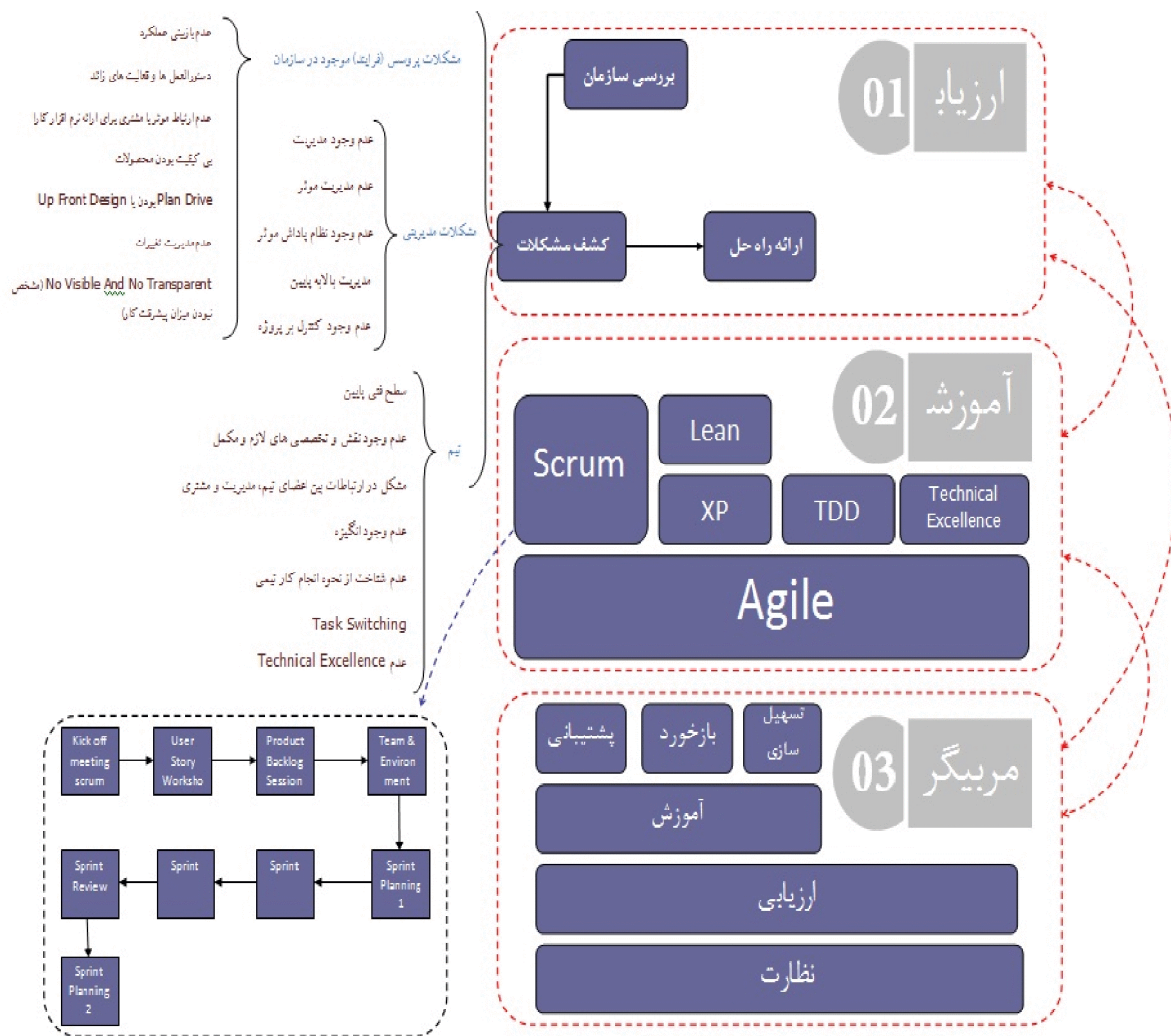
یکی دیگر از افتخارات گروه، برگرداندن بیانیه توسعه نرم افزار چابک به زبان فارسی و انتشار آن در خود سایت بین المللی بیانیه چابک می باشد که از این طریق زبان فارسی و نام کشور ایران نیز در کنار کشورهای دیگر چشم نوازی می کند حتی نام کشور ایران بسیار قبل تر از کشورهای پیشرفته ای مانند ژاپن یا آلمان، ایتالیا و... در لیست قرار گرفته بود. از جمله دیگر عملکردهای غرور آفرین این گروه می توان به برگزاری اولین دوره CSM (Certified Scrum Master) با حضور استاد نامی هنریک کنیبرگ از کشور سوئد و حضور هیئت مدیره تشکل Agile Alliance اشاره کرد. علاوه بر امتیاز اولین دوره بودن این دوره در ایران و کشورهای همجوار، می توان به حمایت مستقیم Scrum Alliance از این دوره نیز اشاره کرد.

هدف IR Agile

چابک سازی شرکت های توسعه نرم افزار با هدف رفع مشکلات و کاستی ها طی مشاوره و آموزش

خدمات قابل ارائه

- آموزش متد های چابک و ارائه خدمات لازم جهت چابک سازی
- آموزش و ارائه مدارک بین المللی Scrum Alliance
- مشاوره در زمینه چابک سازی



برای اطلاعات بیشتر و یا درخواست خدمات می توانید از رسانه های ارتباطی زیر استفاده نمایید :

سایت : [Http://www.iragile.com](http://www.iragile.com)

ایمیل : info@iragile.com

ایمیل : asad.safari@gmail.com

شماره تلفن : 98 0914 912 0914 + (صفری)