

داستان‌های کاربر

توسعه‌دهندگان، ویژگی‌های مورد نظر پروژه را با جمع‌آوری نیازمندی‌ها، در قالب داستانهای کاربر احصاء می‌کنند و به هر کدام متناسب با پیچیدگی‌اش امتیازی اختصاص می‌دهند. با لیستی از داستان‌های دارای ابعادی مشخص و بودجه و زمان مورد نیاز برای هر کدام، مشتریان قادر به این انتخابند که کدام ویژگی‌ها در تکرار (iteration) بعدی باقی بماند. مشخص کردن بودجه و زمان، یعنی تعیین حجم کاری که تیم توسعه برای انجام آن ویژگی، نیاز می‌داند. برآورد بودجه مورد نیاز تکرار اول به صورت تجربی خواهد بود و ممکن است این تخمین در ابتدا نادرست باشد؛ اما با شروع تکرار بعدی درست خواهد شد. در پایان هر تکرار، امتیازات به دست آمده از داستان‌های کامل شده را جمع کنید. مجموع این امتیازات، نشانگر سرعت شما خواهد بود. این سرعت شاخص خوبی جهت چگونگی بودجه‌بندی مرحله بعد است. هنگامیکه امتیازات جمع‌آوری شده به حد مطلوبی رسید، «سرعت پیشروی»، شاخص مناسب دیگری برای بودجه‌بندی است که عبارت است از متوسط سرعت سه تکرار آخر.

با این کار شما به دیدگاه مناسبی از فاز برنامه‌ریزی دست پیدا می‌کنید. حال اجازه دهید نگاه دقیق‌تری به شیوه‌های برنامه‌ریزی داشته باشیم.

برنامه‌ریزی (planning game) دو فاز دارد: فاز شناسایی و فاز برنامه‌ریزی. در فاز شناسایی، توسعه‌دهندگان و مشتریان را دور هم جمع می‌کنند تا درباره نیازمندیهای سیستم در حال طراحی، گفتگو کنند. به خاطر داشته باشید که این کار تا وقتی انجام می‌شود که به ویژگی‌هایی (features) کافی برای شروع انجام کار برسیم و البته واضح است که چنین لیستی از ویژگی‌های احصاء شده، هرچقدر هم که تلاش شود، کامل نخواهد بود. مشتریان اغلب اوقات، خواسته‌ی خود را یا نمی‌دانند یا نمی‌توانند به خوبی توضیح دهند. بنابراین معمولاً این لیست به مرور تغییر می‌کند. در ضمن آنکه برخی ویژگیها دقیق‌تر می‌شود، مواردی نیز ممکن است به لیست افزوده شوند یا حتی می‌توان برخی ویژگی‌های نامربوط را از لیست حذف کرد. در مرحله شناسایی، ویژگی‌ها به داستانهای کاربر تجزیه شد و ثبت می‌شوند.

یک داستان کاربر عبارت است از توصیفی کوتاه از یک ویژگی که نمایانگر یک واحد ارزش کسب و کار برای مشتری است. داستانهای کاربر از زبان کاربر بیان شده‌اند و قالب نوشتاری زیر را دارند:

به عنوان «نوع کاربر»، من می‌خواهم «یک فعل» تا «منفعتی برای کسب و کار»

یا به صورت:

به منظور «یک دلیل» به عنوان «نقش کاربر» من می‌خواهم «یک فعل»

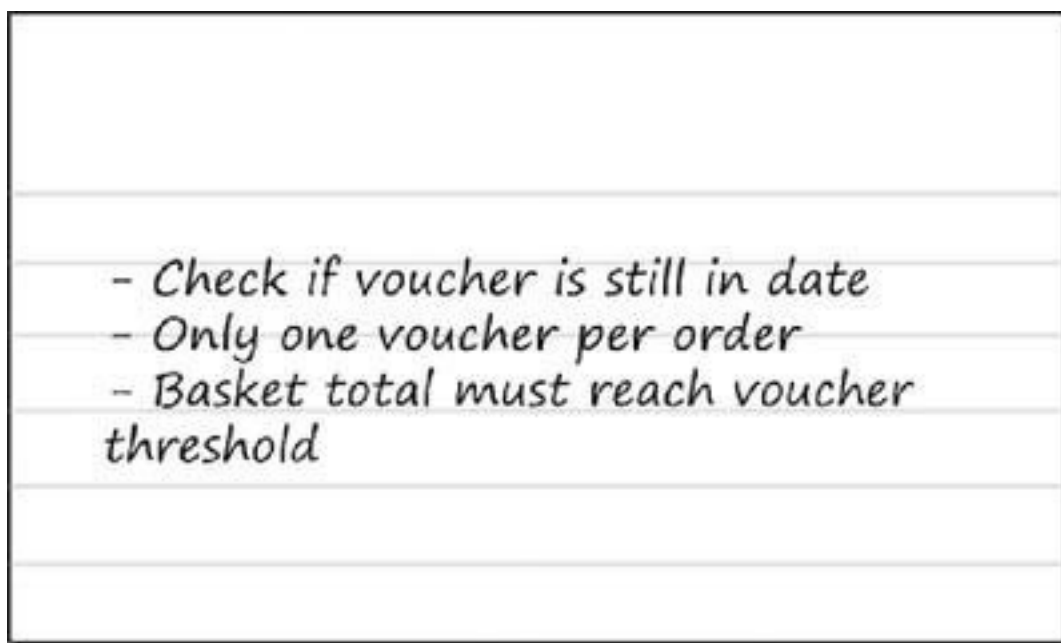
داستانهای کاربر معمولاً در جلسه‌ی گفتگو با مشتری بر روی کارت‌های راهنما نوشته شده و در آن از واژگان و ادبیاتی استفاده می‌شود که برای مشتری قابل فهم باشد. ممکن است چنین بیانیدیشید که ثبت نیازمندی‌ها، خلاف مزیت‌های چابک‌سازی است؛ چرا که تولید نرم‌افزار کارآمد و چابک مبتنی بر مستندسازی گسترده و فراگیر خواهد بود. در واقع، داستان‌های کاربر به طور ساده فقط یادآورنده جزئیات بیشتری از گفتگوی انجام شده‌اند که به عمد به صورت کوتاه و دقیق نوشته شده‌اند. فهم دقیق‌تر جزئیات کار، مستلزم ارتباط بیشتر میان توسعه‌دهندگان و مشتری است. در واقع همسو با این اصل چابک که می‌گوید: «مؤثرترین و کارآمدترین شیوه انتقال اطلاعات در میان تیم توسعه و به خارج از آن، گفتگوی چهره به چهره است.»

هنگام احصاء ویژگی‌های پروژه تحت عنوان داستان‌های کاربری، از اصول INVEST (که پیش‌تر گفته شد) جهت کنترل مناسب بودن این داستانها استفاده کنید. شکل 2-3 مثالی از یک داستان کاربر را که توصیف‌کننده ویژگی «افزودن یک بن تخفیف به سبد خرید»

است، نشان می‌دهد. «تخفیف گرفتن»، یک منفعت کسب و کار است برای عامل (actor) اصلی، یعنی مشتری. «یک بن تخفیف به سبد بیفزاید» نام فرآیند یا «use case» مربوط است.



از معیار پذیرش (acceptance criteria) نیز می‌توان در هنگام تولید داستان‌ها استفاده کرد. معیار پذیرش را می‌توان در پشت کارت داستان، آن طور که در شکل 3-3 نشان داده شده است، نوشت. استفاده از طرف مقابل کارت این اجازه را می‌دهد که اعضای تیم و مشتریان، اطلاعات خودشان را در یک جا جمع کنند.



معیار پذیرش همچنین به تشخیص جزئیات بیشتر یا شناسایی وابستگی‌ها کمک می‌کند. مثلاً در شکل 3-3 تعریف «in date» چیست و چه چیزی حدود یک بن تخفیف را مشخص می‌کند؟ معمولاً باید حداقل سه معیار پذیرش وجود داشته باشد. در فصل بعد در یک مطالعه موردی، مطالب بیشتری را دربارهٔ داستانهای کاربر خواهید آموخت.

هنگامیکه تیم و مشتریان حس کنند که حدود 75 درصد از ویژگی‌های اصلی احصاء شده است، توسعه‌دهندگان ابعاد داستان‌ها را تخمین زده و آنها را برای اولویت‌بندی توسط مشتری آماده می‌کنند.

تخمین

شکی در آن نیست که تخمین‌زدن کار سختی است. تخمین‌زدن هم دانش است هم هنر. تخمین‌زدن در یک پروژه تازه شروع شده، بسیار سخت است زیرا مجهولات بسیاری در آن وجود دارد.

یکی از روش‌های تخمین گروهی، روش «Planning Poker» نام دارد. در این روش همه‌ی اعضای فنی تیم، متشکل از توسعه‌دهندگان نرم‌افزار، تحلیل‌گران، متخصصان امنیت و زیرساخت، مشارکت می‌کنند. نقش مشتری در این حالت پاسخ‌گویی به سؤالات احتمالی اعضای تیم است تا ایشان بهتر بتوانند تخمین بزنند.

شیوهٔ انجام کار به این صورت است که عضوی از تیم، یک داستان کاربر را برداشته و آن را برای تیم توضیح می‌دهد. تیم دربارهٔ آن ویژگی با مشتری گفتگو کرده تا جزئیات بیشتری را دریابد. وقتی که تیم به درک خوبی از آن رسید، رأی‌گیری آغاز می‌شود. هر عضو تیم با یک کارت، از مجموعه‌ای از کارتهایی با شماره‌های 0، 1، 2، 3، 5، 8، 13، 20، 40 و 100 رأی خود را اعلام می‌کند.

تیم باید از داستانی شروع کند که نسبتاً کوچک و ساده باشد. این داستان به عنوان مبنا انتخاب می‌شود. هر تخمین داستان کاربر، باید به نسبت این داستان کوچک انجام شود. اگر داستان مبنا به خوبی انتخاب نشود، بقیهٔ تخمین‌ها نادرست خواهد بود.

اگر همه‌ی اعضای تیم به یک صورت رأی دهند، آن رأی، تخمین آن داستان خواهد شد. اگر اختلاف آراء وجود داشت، ناظر یعنی کسی که رأی نمی‌دهد، از افرادی که بالاترین و پایین‌ترین امتیاز را داده‌اند، می‌خواهد که علل خود را توضیح دهند. سپس تیم مجدداً گفتگو کرده و دوباره رأی‌گیری می‌کند. طبق تجربه، خوب است که زمان معقولی، برای هر گفتگو در نظر گرفته شود.

اگر تخمین یک داستان به دلیل فقدان دانش فنی، بسیار سخت بود، مناسب است که این داستان کنار گذاشته شود و داستان دیگری برای برطرف کردن مشکل ناآشنایی با دانش فنی مورد نظر فراهم شود. بدین ترتیب تیم توسعه در موقعیت بهتری می‌تواند نسبت به داستان جدید تخمین بزند.

داستان‌هایی که بیش از یک هفته کار نیاز داشته باشند با عنوان داستانهای حماسی (epic stories) شناخته می‌شوند و معمولاً برای تخمین بسیار بزرگ هستند. در واقع، این داستان‌ها به چند داستان کوچک‌تر که قابل فهم‌تر و به آسانی قابل تخمین باشند، تجزیه می‌شوند. این بدان معناست که ایجاد یک داستان کاربر از تعداد انبوهی ویژگی موجب کاهش کارایی خواهد شد.

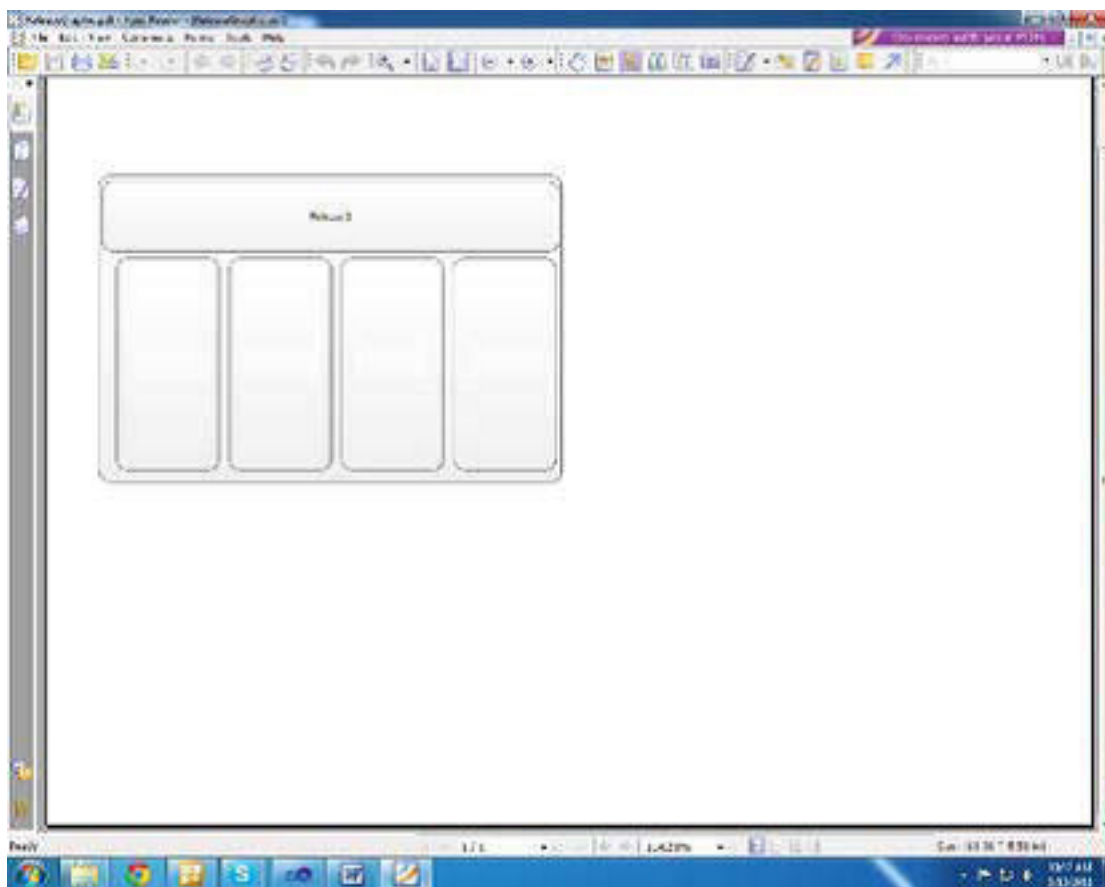
تخمین در تیمی که افراد آن تاکنون با همدیگر سابقهٔ همکاری نداشته باشند، خیلی پایین یا خیلی بالاست. اما با استمرار هر تکرار و تجربه و دانش بیشتر افراد، تخمین داستان‌ها بهتر می‌شود.

استفاده از ابزار Planning Poker مزایای بسیاری دربردارد. دقت تخمین بالا می‌رود؛ زیرا مسأله از منظر تخصص‌های گوناگون مورد بررسی قرار گرفته است. همچنین به تیم کمک می‌کند که هم رأی شوند و گفتگو میان اعضا را تسهیل می‌کند. پس از آنکه داستان‌ها تخمین زده شدند، مشتری و صاحب محصول با تیم توسعه در تولید چگونگی انتشار نسخه‌ها، همکاری می‌کنند.

برنامه انتشار

اگرچه کدهای قابل ارسال، قابلیت انتشار در پایان هر تکرار را دارند، اما یک پروژه XP در چند سری منتشر شده است. یک نسخهٔ منتشرشده، متشکل از تعداد مناسبی داستان برای عرضهٔ ارزش کسب وکاری است که به کوچک نگه داشتن آن کمک می‌کند.

بسیار مناسب است که یک موضوع یا هدف خاص را در ضمن هر نسخه انتشار، مد نظر قرار داد تا کمک کند که هر نسخه انتشار بر برخی ارزشهای کسب و کاری متمرکز شده و آن را هدایت کند. معمولاً یک نسخه انتشار، متشکل از چهار تکرار است؛ همانطور که در شکل 3-4 نشان داده شده است.



در برنامه ریزی نسخه های انتشار، طول یک تکرار نیز تعیین می شود که معمولاً بین دو تا چهار هفته است. مطابق تجربه، اگر محیط کار شما دچار بی نظمی و اختلالات دائمی است، می توانید دوره تکرار را به یک هفته محدود کنید.

یکی از پروژه هایی که ما بر روی آن کار می کردیم، برنامه ای بود که نگهداری آن بسیار سخت و فوق العاده ناپایدار بود. مشتری مکرراً با تیم تماس گرفته و اشکالات بحران ساز و ایراداتی را که مخل برنامه بودند، گزارش می کرد. در ابتدای کار دوره، تکرار ما هفتگی بود. به همین دلیل چون حلقه بازخوردگیری مان کوچک بود، می توانستیم بر پایدارسازی پروژه در هر دوره کاری تمرکز کنیم. هنگامی که محصول به پایداری مناسب تری رسید و تماس های مشتری کم شد، قادر شدیم تا در هر دوره، دقت بیشتری بر روی مسائل به خرج دهیم.

اگر قصد دارید به صورت دقیق بر روی حلقه بازخورد متمرکز شوید، دوره ی تکرار یک هفته ای، مدل خوبی است. اما این مدل سربرای زیادی را به دلیل ضرورت تقسیم داستانهای کاربر باید به بخش های کوچک تری تا آن اندازه که در یک دوره تکمیل شوند، بر پروژه تحمیل می کند. در ادامه خواهیم گفت که هر تکرار شامل برنامه ملاقات و بازبینی نیز هست.

بعد از مدتی که تیم با فرآیند کار آشنا تر شد و نوبت به مشکلات با اولویت کم تر رسید، می توان دوره تکرار را دو هفته ای در نظر گرفت. اما اگر پروژه به گونه ای است که ویژگی های بزرگ تر را نمی توان به موارد کوچک تری که قابل انجام در دوره های یک هفته ای باشد، تجزیه کرد و تیم هنوز در حال یادگیری است، دوره های بلند مدت تر قابل پذیرش است.

مشتری با توجه به طول دوره تکرار و بودجه داستان آغازین، انتخاب می‌کند که کدام داستان در هنگام انتشار نسخه اول، در تکرار اول کامل شود.

این مشتری است که داستان‌ها را به گونه‌ای اولویت‌بندی می‌کند تا مشخص شود که کدام یک بیشترین ارزش کسب و کار را فراهم می‌کند. از آنجایی که مشتری مسئول داستانهای کاربر است، تیم باید به وی توضیح دهد که داستانهایی وجود دارند که صرفاً باید به جهت دلایل فنی ایجاد شوند.

معمولاً باید به داستانهای کاربری‌ای که مستلزم ریسک بالا بوده یا دربرگیرنده مجهولات زیادی باشند، بیش از یک یا دو تکرار اختصاص داد.

برنامه تکرار

مشتری داستان‌هایی را که می‌خواهد در تکرار باشند، انتخاب می‌کند. برای هر داستان کاربر، مجموعه‌ای از معیارهای پذیرش، تعریف شده است. همان طور که متوجه شده‌اید ما در هر فاز، وقت بیشتر و بیشتری را صرف جمع‌آوری جزئیات هر داستان کاربر کرده و بصورت عمیق‌تری در آن غور می‌کنیم. این کار مفید است، زیرا اگر یک داستان کاربر ایجاد شده در ابتدای پروژه، ممکن است بعداً به عنوان داستانی کم اهمیت یا غیر مهم دیده‌شود و بدون آنکه وقت خاصی برای آن صرف شده باشد، کنار گذاشته شود. اما اگر در ابتدای کار وقت زیادی صرف دقیق‌تر کردن داستان‌های کاربر شود و بعداً بعضی از آنها کنار گذاشته شوند، در واقع وقت تلف شده است. بنابراین دقیق‌تر کردن یک داستان در جایی که مورد نیاز است، باید اتفاق بیفتد. در سطح برنامه تکرار، مجموعه‌ای از معیارهای پذیرش را برای هر داستان کاربر تعریف می‌کنیم. معیار پذیرش به توسعه‌دهنده کمک می‌کند تا بداند که یک داستان کاربر به طور کامل انجام می‌شود. این معیارها به صورت مؤلفه‌هایی از بافرض/هنگامی که/درنتیجه، نوشته می‌شود.

مثالهای زیر چگونگی انجام این کار را توصیف می‌کند:

عنوان ویژگی : افزودن کالایی به سبد

به عنوان یک مشتری می‌خواهم بتوانم کالایی را به سبدم اضافه کنم؛ به نحوی که قادر باشم به خرید خود ادامه دهم.

سناریو : سبد خالی

با فرض اینکه یک سبد خالی دارم، در نتیجه جمع تعداد کالایی که برای سفارش در سبد من وجود دارد، صفر است.

سناریو : افزودن یک کالا به سبد

با فرض اینکه یک سبد خالی دارم هنگامی که کالایی با شناسه 1 به سبدم اضافه می‌کنم، در نتیجه جمع کالاهای قابل سفارش در سبد 1 می‌شود.

سناریو : افزودن کالاهایی به سبد

با فرض اینکه یک سبد خالی دارم، هنگامی که کالایی با شناسه 1 و کالایی با شناسه 2 به سبدم اضافه می‌کنم، در نتیجه جمع کالاهای قابل سفارش در سبد 2 می‌شود.

سناریو : دو بار افزودن یک کالا

با فرض اینکه یک سبد خالی دارم هنگامی که کالایی با شناسه 1 به سبدم اضافه می‌کنم و هنگامی که کالایی با شناسه 1 را مجدداً به سبدم اضافه می‌کنم، در نتیجه تعداد کالاهای با شناسه 1 در سبد من باید 2 باشد.

سناریو : افزودن یک کالای تمام شده به سبد

با فرض اینکه یک سبد خالی داریم و کالایی با شناسه 2 در انبار وجود نداشته باشد، هنگامی که من کالایی با شناسه 2 را به سبد خودم اضافه می‌کنم، در نتیجه جمع تعداد کالای قابل سفارش در سبد من باید 0 باشد و به کاربر، موجود نبودن آن کالا را هشدار دهد.

یک آزمون پذیرش (acceptance) به زبان متعارف در قوانین کسب و کار نوشته می‌شود. در مثال سبد خرید، این سؤال پیش می‌آید که چگونه می‌توان یک محصول را از سبد کالا، حذف کرد و اگر یک جنس اکنون در انبار نیست و کاربر پیام هشدار دریافت کرده است، در ادامه چه اتفاقی باید بیفتد؟ سناریوها به تیم در کشف ملزومات کسب و کار و تصریح آن‌ها کمک می‌کند.

این سناریوها توسط توسعه‌دهنده به عنوان نقطه شروع آزمونهای واحد در توسعه آزمون محور و رفتار محور استفاده می‌شود. سناریوها همچنین در آزمون معیارهای پذیرش به توسعه‌دهنده کمک کرده و توسعه‌دهنده و تست‌کننده را قادر می‌سازند که بر روی اتمام داستان اتفاق نظر داشته باشند.

بعد از آنکه سناریوهای معیار پذیرش تعیین شد، تیم توسعه، هر داستان را به تعدادی وظیفه تقسیم می‌کند و وظایف مرتبط به یک داستان، در تابلوی وظایف قرار گرفته و تیم توسعه تخمین‌های خود را در قالب یکی از واحدهای اندازه‌گیری، مثلاً نفرساعت اعلام می‌کند. شکل 3-5 یک تابلوی وظیفه را نمایش می‌دهد.

به عنوان مثال وظایف می‌توانند شامل ایجاد طرح یک بانک اطلاعاتی برای یک داستان یا یکپارچه‌سازی آن با بخشی موجود در سیستم باشند. وظایف شامل مؤلفه‌های فنی مانند تهیه گزارش از زیرسیستم‌ها یا چارچوب مدیریت استثنائات نیز می‌باشد. اغلب این‌گونه وظایف نادیده گرفته می‌شود. یک داستان کاربر با وظایف گوناگونی گره خورده است. مثلاً:

داستان کاربر : به عنوان یک کاربر می‌خواهم بتوانیم یک کاربر را مدیریت کنیم.

وظایف زیر از این داستان قابل استخراج است:

طرحی برای بانک اطلاعات جهت ذخیره‌سازی اطلاعات کاربر ایجاد کن.

یک کلاس کاربر، برای مدیریت کاربر از درون برنامه ایجاد کن.

هر عضو تیم می‌تواند بر روی هر وظیفه‌ای که بر روی تخته است، کار کند. هنگامیکه یک عضو گروه، وظیفه‌ای را برمی‌دارد، باید نشانی از خود روی کارت آن وظیفه قرار دهد (مثلاً حروف اوّل اسمش) تا بقیه افراد بدانند که وی بر روی آن وظیفه، مشغول به کار است. معمولاً اما نه همیشه، یک توسعه‌دهنده همه وظایف مربوط به یک داستان را برمی‌دارد. این کار بدین معناست که آن توسعه‌دهنده با پشتیبانی تیم، مسئول اتمام آن کار است.

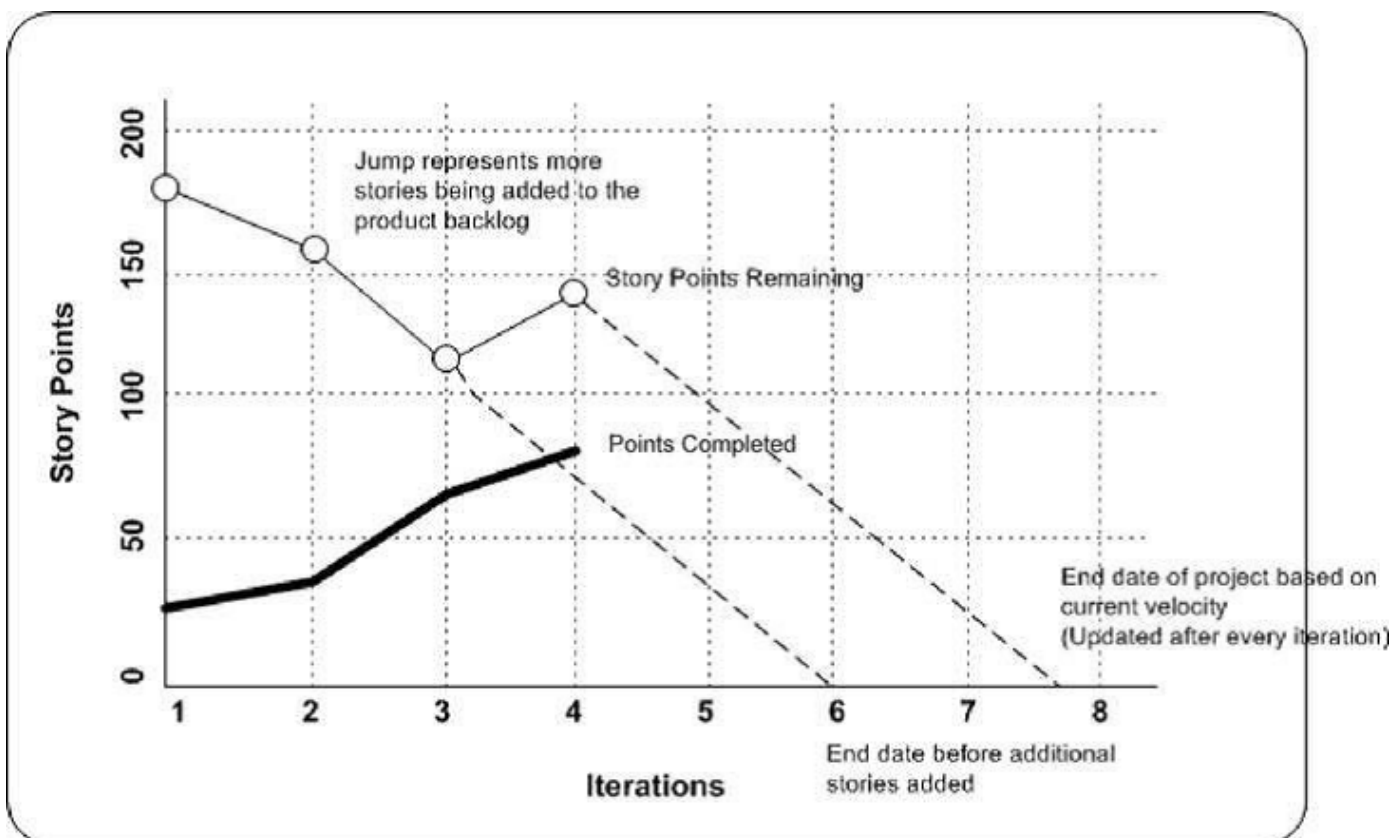
Stories	Task Back log		In Process	In Testing	Done
Basket Discount Vouchers In order to receive a discount on my order As a Customer I want to be able to add a voucher to my basket	Asadiklj kj l ljljla oijlk Uoiuola jkondfghalf kjkd/ dfgfdg n khskjdghfjkhsdj safadkln lkjh lll jwz lkjkl sfjnsadlk jljayk k ;	Asadiklj kj l ljljla oijlk Uoiuola jkondfghalf kjkd/ dfgfdg n khskjdghfjkhsdj safadkln lkjh lll jwz lkjkl sfjnsadlk jljayk k ;	Asadiklj kj l ljljla oijlk Uoiuola jkondfghalf kjkd/ dfgfdg n khskjdghfjkhsdj safadkln lkjh lll jwz lkjkl sfjnsadlk jljayk k ;		
Loyalty Points In order to receive free gifts As a Customer I want loyalty points for orders I place	Asadiklj kj l ljljla oijlk Uoiuola jkondfghalf kjkd/ dfgfdg n khskjdghfjkhsdj safadkln lkjh lll jwz lkjkl sfjnsadlk jljayk k ;	Asadiklj kj l ljljla oijlk Uoiuola jkondfghalf kjkd/ dfgfdg n khskjdghfjkhsdj safadkln lkjh lll jwz lkjkl sfjnsadlk jljayk k ;		Asadiklj kj l ljljla oijlk Uoiuola jkondfghalf kjkd/ dfgfdg n khskjdghfjkhsdj safadkln lkjh lll jwz lkjkl sfjnsadlk jljayk k ;	Asadiklj kj l ljljla oijlk Uoiuola jkondfghalf kjkd/ dfgfdg n khskjdghfjkhsdj safadkln lkjh lll jwz lkjkl sfjnsadlk jljayk k ;
	Asadiklj kj l ljljla oijlk Uoiuola jkondfghalf kjkd/ dfgfdg n khskjdghfjkhsdj safadkln lkjh lll jwz lkjkl sfjnsadlk jljayk k ;				

به محض اینکه یک تکرار آغاز شد، داستان‌هایی را که کار بر روی آنها شروع شده است، دیگر نمی‌توان تغییر داد. این مهم است که برنامه تکرار را در حین انجام آن، تغییر ندهید؛ زیرا این کار منجر به سوئیچینگ زمینه (context switching) می‌شود. برای توسعه‌دهندگان سوئیچینگ زمینه، هم به لحاظ زمانی و هم به لحاظ مالی، بسیار پرهزینه است.

به جای آنکه تلاش کنید در ضمن یک تکرار، تغییراتی را ایجاد کنید، مشخص کنید که آیا این کار اضافه، یا داستان اضافه را می‌توان تا تکرار بعدی به تعویق انداخت. مشتریان یا مدیران معمولاً می‌توانند چنین تعویقی را بپذیرند؛ زیرا این پذیرش مستلزم به تأخیر انداختن کار، مثلاً تا یک ماه دیگر نیست. اگر این کار جدید را که اضافه شده است، نمی‌توان به تعویق انداخت، باید ریسک خارج ساختن کدهای موجود و رفتن به سمت کدنویسی برای کارکرد جدید را به همراه تیم بررسی کرد. همچنین تیم باید بداند که اگر کار اضافه‌ای به یک دوره تکرار افزوده شد، بخشی از کارهای این دوره باید به تکرار بعدی موکول شوند. قاعده کلی این است که اگر چیزهای جدیدی به کار وارد شد و تعویق آن ممکن نبود، باید کارهایی با همان ابعاد یا بزرگتر از تکرار، خارج شود.

سرعت به ما نشان می‌دهد که تیم چه حجم کاری را در طول یک دوره کامل کرده است. از سرعت، در برنامه‌ریزی تکرارهای آتی استفاده می‌شود. برای درک چگونگی سرعت کار از نمودار burn-down استفاده می‌شود. یک نمودار burn-down (شکل 3-6) داستان‌های باقی‌مانده یک پروژه و داستانهای تکمیل شده را در یک تکرار نمایش می‌دهد. سرعت در پایان هر تکرار محاسبه می‌شود و تعریف آن عبارت است از تعداد داستانهای تکمیل شده در آخرین تکرار. بر اساس سرعت کنونی و تعداد داستانهای باقی‌مانده، می‌توان تخمین زد که چقدر طول می‌کشد تا همه‌ی داستانها تکمیل شود. همانند آنچه در شکل 3-6 با خط چین نمایش داده شده است.

نمودار burn-down ابزار خوبی برای فهم آن است که آیا تیم می‌تواند پروژه را در زمان مقتضی به پایان برساند یا خیر و اگر نمی‌تواند، مدیر چگونه باید نسبت به آن تصمیم‌گیری کند. آیا افراد بیشتری باید به پروژه وارد شوند؟ آیا باید از ویژگی‌های مدنظر پروژه کاهش داد، یا باید زمان پایان کار را تغییر داد؟



در طول یک تکرار، هر روز باید گفتگوهای سرپایی با حضور همه اعضای تیم انجام شود و مشکلاتی که ممکن است باعث به تأخیر افتادن ارائه کار شود، مورد بحث و بررسی قرار گیرد و همچنین تیم، لیست وظایف و تخته آن را به روز کرده تا پیشرفت یا موانع آن به وضوح قابل رؤیت باشند.

با تشکر از آقای [سید مجتبی حسینی](#)