

Case study: Distributed Scrum Project for Dutch Railways

Dutch railways - IS

- Přeprava 1,2 milionu lidí denně
- Nový informační systém
 - Více přesný
 - Více automatický
 - Více informací
- Součást systému – **Publish system (PUB)**

Publish system - (PUB)

- 1. pokus tradiční přístup – vodopád
 - Zrušen po 3 letech
 - Dodavatel nedokázal dokončit funkční systém
- 2. pokus zavedení agilního přístupu – SCRUM
 - Vytváření systému od začátku
 - Zaměřeno
 - Na kooperaci se zákazníkem
 - Otevřenou diskusi
 - Práce po menších inkrementech

Počátek procesu

- Problém najít správného „Product Owner“
 - Nominování 2 business analytikové
 - Zapojení do předchozího pokusu
- Důležitá rozhodnutí – Project manager
 - Méně času
 - Méně know-how o požadavcích
- Výsledkem – Product owner – více lidí

Počátek procesu

- Software
 - Součást většího souvisejícího software
 - Nutnost poskytnutí přehledů plánování
 - Deadline meeting klíčové pro organizování
 - Nějaká estimace lepší než žádná estimace
- 7 lidí v týmu – 2 týdenní sprinty
- Plánované navýšení o lidi z Indie

Počátek procesu

- Setkání všech členů týmu
 - Dohodnutí pracovních postupů
 - Používání nástrojů
 - Core hours
- Zaznamenání setkání na wiki stránku – postupné updaty

Vývoj software

- První iterace
 - Vyrobení, testování a ukázka klíčových user stories
 - Uspokojení zákazníka – dříve, lépe, lepší kontrola než u předchozího pokusu
- Po několika iteracích navýšení zdrojů – lidí
 - 2 týmy (5 developerů) + 1 tester
 - Později 3 týmy + 3 testeři
 - Komunikace mezi Nizozemskem a Indií - Skype

Testování

- Aplikovány automatizované testy kvůli kontrole na konci každého sprintu
- přes nárůst systému, pouze jeden tester na 8 členný tým
- 2 stupně testování
 - Unit testy
 - Akceptační testy

Testování

- Akceptační testy automatizovány použitím FitNesse
- Každá user story -> akceptační testy FitNesse
- Výhoda – vytváření testů před kompletní implementací user story

Testování

- Složité testování UI
 - Spoléhání na manuální zkoušení
 - Narůst prvků => časově složitější
 - Nacházení zásadních bugů pouze v této oblasti
 - Poučení
 - Přes složitější stavbu testů se u složitých systému vyplatí je psát než spoléhat na manuální testování
 - Později se to projeví a vyplatí

Výsledek

- Velmi spokojený zákazník
- Jeden z hlavních faktorů úspěchu projektu
 - Pravidelná diskuse se zákazníkem při vývoji software
- 20 man-years
- 100 000+ řádků kódu

Shrnutí

- It can be hard to find a product owner with both detailed knowledge of the requirements as well as the mandate to set priorities. Often it is unavoidable to fill the product owner role with multiple people, especially in large projects.
- When meeting a deadline is important, it's important to make sure that the product backlog is complete and estimated. For requirements, any estimate is better than no estimate, even if little information is available. In combination with the team's velocity, this provides the necessary information for release planning.
- Scrum is well-suited to execution with multiple distributed teams. Having each Scrum team contain resources both in the Netherlands and India was good for team spirit and forced us to work on effective communication. For communication, off-the-shelf hardware and free software make the cost of implementing this low.

Shrnutí

- It is useful to start a distributed project with an initial co-located session to reach agreement on team practices.
- Work that doesn't fit well in a Scrum Sprint (i.e. chasing down key people, interfacing with other customer departments) can be handled more effectively by a separate team. This allows the feature teams to focus on building the software. Using a dedicated technical writer also helps in this respect, even if it does add communication overhead.
- Although not needed for the software development process, extensive requirements documentation may still be required by the customer. In a Scrum project, however, this cannot replace the use of user stories. If both are used, the overhead of reconciling requirements in two places should be factored into planning.
- Automated testing is vital to deliver software incrementally, unhindered by regression bugs. Before the project is over, the return on investment will outweigh the cost.

Děkuji za pozornost