

Retour d'expérience Atchik- Realtime

27/03/2009

Jean-Baptiste Chéry & Pascal Rieux (ScrumMasters)



Atchik-Realtime

- Plus de 10 ans d'expérience dans le développement de services pour la téléphonie mobile. 110 employés avec des bureaux en France, au Danemark, au Brésil, en Colombie et au Mexique
- Services déployés dans plus de 40 pays, plus de 250 million d'abonnés utilisent des services déployés dans plus de 40 pays en Europe, Amérique, Afrique et au Moyen-Orient
- Quelques références:



Algeria
Botswana
Brazil
Cameroon
Chile
Colombia
Denmark
Ecuador
France
Ghana
Guatemala
Honduras
Ivory Coast
Jamaica
Jordan
Madagascar
Mali
Mexico
Morocco
Nicaragua
Oman
Panama
Peru
Poland
Portugal
Puerto Rico
Romania
Senegal
Spain
Switzerland
Venezuela

Les applications

- Service de Chat et de flirt sur mobile + web
- Gestion d'évènements
- Outils de modération
- Outils d'animation



L'environnement technique

- Développements java / mysql – oracle
- Plateformes de production :
 - Cluster Sun ~15 serveurs en mode SaaS
 - Cluster linux 9 serveurs pour Orange France (plateforme dédiée). Plus de 6 millions de hits HTTP/jour
 - Cluster linux 4 serveurs pour Orange Roumanie
- Plateformes de pre-production (SaaS/Orange)
- Plateformes d'intégration

Les outils

- Gestion des backlogs, des releases, des sprints, des tests, des équipes, reporting : Rally
- IDE : eclipse (majoritairement)
- Build : maven 2
- Repository : archiva (apache)
- Contrôle de version : subversion
- Intégration continue : continuum (apache)
- Bug tracking : bugzilla
- Documentation
 - Interne : wiki (changelogs, bilan de spikes, notes techniques, bilan de sprints)
 - Externe : doc

Avant scrum

- Cycle en V
- Problèmes classiques liés à cette méthodologie :
 - Peu de réactivité par rapport aux demandes d'évolutions tardives
 - Difficultés à planifier une date réaliste de fin de projet pour un périmètre fonctionnel fixe
 - Long travail de spécification (tentative de « verrouillage » des demandes d'évolution)
 - Peu ou pas d'interactions formalisées client-technique pendant la réalisation

Chronologie

- Premières expérimentations SCRUM fin 2007 (département technique CPH)
- SCRUM fut ensuite préconisé par le management pour l'ensemble des développements
- Formations + coaching scrum : en avril et mai 2008 (TLS), fin 2008, début 2009 (CPH)
- Début du 1er sprint : juin 2008 (TLS) et janvier 2009 (CPH)

Les équipes actuelles

- Copenhague : 2 équipes
 - 1 équipe « service »
 - 2 product owners
 - 1 scrum master
 - 4 développeurs
 - 1 équipe « système » partiellement « scrumée »
 - 4 membres + 1 product-owner
- Toulouse : 2 équipes (par ligne de produit)
 - 2 product owners
 - 2 ScrumMasters/développeurs
 - 4 développeurs (2+2)
 - 2 testeurs
- Les product-owners interviennent localement
- Pas d'équipe multi-site (sauf pour certains tests fonctionnels)

L'environnement de travail

- Espace ouverts
- Regroupements par « îlot » suivant les projets (un îlot = 4 postes)
- Un îlot regroupe soit des développeurs, soit des développeurs et des testeurs
- Possibilité de « pair-programming »
- Un tableau par équipe

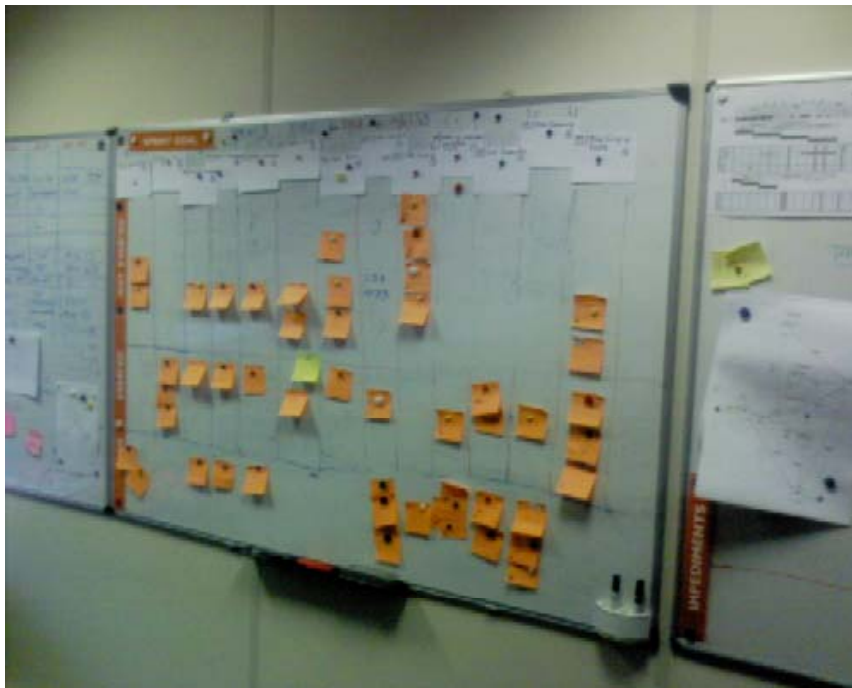


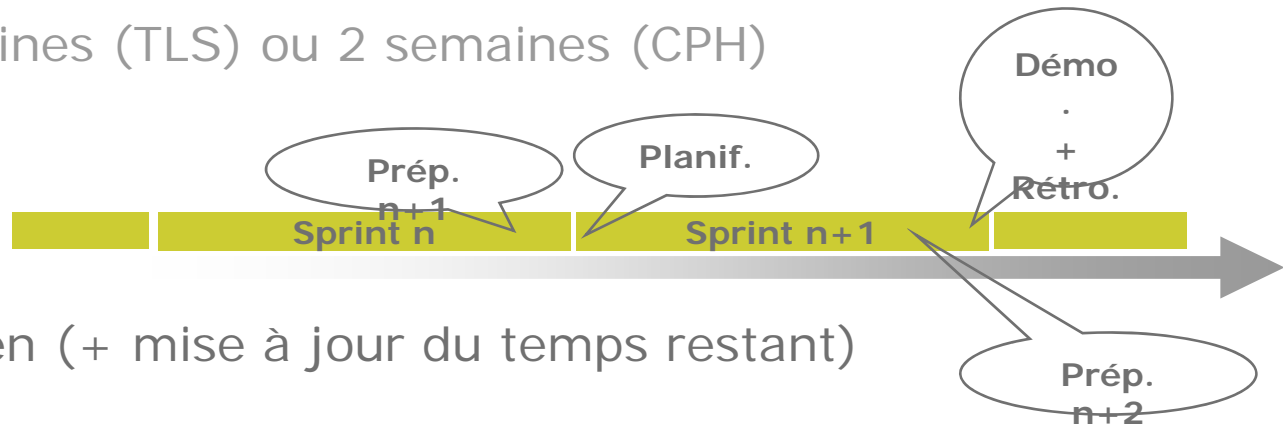
Tableau « standard »



Une variante...

Les pratiques

- En amont: élaboration de la roadmap (dept marketing + technique)
- Sprints de 3 semaines (TLS) ou 2 semaines (CPH)
- Réunions :
 - Préparation
 - Planification
 - Scrum quotidien (+ mise à jour du temps restant)
 - Démonstration
 - Rétrospective
- TDD (pour les tests manuels) à Toulouse (2 testeurs), pas de testeur à CPH => pas de TDD au sens des tests manuels. Sur les deux sites : tests unitaires
- Pair-programming (au cas par cas)

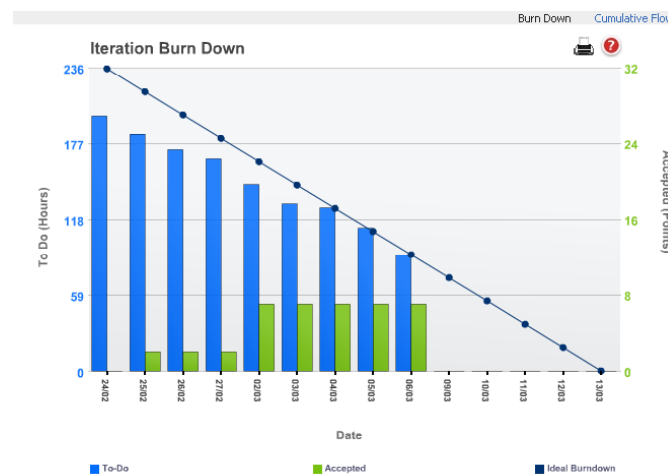
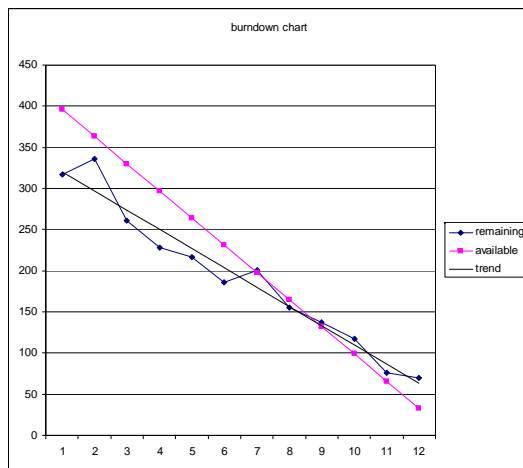


Les pratiques (suite)

- Préparation d'une release
 - Pilotée prioritairement par le marketing en fonction des besoins commerciaux
 - Mise à jour du backlog de release
 - Actée lors de comités opérationnels (management)
- Préparation d'un sprint
 - Revue du backlog de release et des priorités
 - Réalisation des estimations manquantes et nécessaires pour le sprint (en points)
 - Mise à jour du backlog de sprint
- Planification d'un sprint
 - Préalable : calcul de la bande-passante (~6 h/j/personne)
 - Découpage en tâches (post-its)
 - Estimation initiale de la durée de chaque tâche
 - Saisie des tâches et de leur durée dans l'outil
 - Tableau pour la définition de fini (définition adaptée par US)
 - => engagement !

Les pratiques (suite)

- Durant le sprint :
 - User stories, tâches et « defects » dans l'outil
 - Tableau des tâches en //
 - Les post-its mentionnent l'estimation initiale de la durée d'une tâche et le trigramme de celui qui « prend » la tâche
 - Burn-down de sprint revu avant chaque scrum quotidien dans l'outil => mise à jour effectuée avant le scrum quotidien (ajout au rituel : « le reste à faire est à jour »)



Les pratiques (suite)

- Le scrum quotidien
 - Durée : 15 minutes, mais pas de contrôle strict
 - Par le passé, des difficultés pour se limiter au rituel scrum et à la durée de 15 minutes
 - => décision de poursuivre le scrum quotidien par un point technique au besoin
- La démonstration
 - Grande salle de réunion, vidéoprojecteur
 - Ouverte à un large public
 - Le résultat des spikes a été résumé sur le wiki, il est présenté lors de la démonstration
- La rétrospective
 - Revue des points marquants : vélocité réelle/estimée, événements (ajout/retrait de US, blocages)
 - Remontée des points positifs/négatifs avec regroupement
 - => Actions à initier/prolonger/modifier/arrêter

Quelques indicateurs

- Backlog (global, courant) :
 - 80 user-stories planifiées -> juin 2009
 - 83 user-stories non-planifiées
- ~80% des US ont été acceptées depuis juin 2008 (TLS) :

TLS				TLS SaaS team				TLS scrum team			
sprint	estimation	actual	% OK	sprint	estimation	actual	% OK	sprint	estimation	actual	% OK
R5S1	135	112	83%								
R5S2	165	153	93%								
R5S3	139	126	91%								
S8	105	85	81%								
S9	120	112	93%								
S10	80	67	84%								
				DMS1				S11	41	27	66%
				DMS2	49	34	69%	S12	24	9	38%
				DMS3	47	42	89%	S13	30	30	100%
DMS4+OFR	117										
average		109	87%			38	79%			22	68%

Amélioration : la définition de fini

- D'une liste de critères au « contrat » du sprint
 - Lors de la planification : définition des critères applicables US par US
 - Dès que tous les critères sont remplis, la US est acceptable par le PO

Always done	Statistics indicators logged
	Language file (keys are constants)
	Standard packaging
	Installation on test server
	Functional test cases written
	Functional test cases done and ok
Not always done	Jmeter scripts written
	Javadoc source written
	Database requests reviewed (indexes, performances)
not done	UML diagrams
	Release note written
	Unit testing (junit)
	Javadoc package written/updated
	Logs properly written and in proper channel (debug, err, warn, ...)
	Performance individual source
Load tests	
Monitoring facilities	



	US193	US272	US275	US399	US400
Statistics indicators logged	X	X			
Language file (keys are constants)	X				
Standard packaging	X	later	later	later	
Installation on test server	X				
Functional test cases written		X			
Functional test cases done and ok	X	X			
Jmeter scripts written					X
Javadoc source written		X	X	X	
Database requests reviewed (indexes, performances)					X
UML diagrams					
Release note written	X	X	N/A	N/A	
Unit testing (junit)		X	X	X	
Javadoc package written/updated					
Logs properly written and in proper channel (debug, err, warn, ...)	X	X	X	X	
Performance individual source					
Load tests					X
Monitoring facilities					
Tech description written			X	X	X
Findbugs/PMD	X	X	X	X	

Autres axes d'amélioration

- En cours
 - Définition de fini
 - Indicateurs de qualité (code, tests)
 - Automatisation des tests, y compris des tests d'intégration
 - Intégration continue
- Potentiels
 - Définir une valeur pour chaque user-story
 - Effectuer une revue plus formelle
 - Tableau niko-niko (appliqué à CPH)
 - Scrums de scrums
 - Réserver formellement du temps à la veille technologique

Les problèmes

- Tâches non liées à des user-stories
- Traitement de l'historique des bugs
- US vs. spike ?
- « Pouvoir » des product-owners et scrum-masters à isoler les équipes des perturbations
- Généraliser scrum aux équipes non dev. ?
- Pas d'implication des clients « externes » dans le processus

Conclusion

- Les équipes ont parfaitement et rapidement intégré les concepts de l'agilité
- Scrum a renforcé l'esprit d'équipe (point remonté à chaque retrospective)
- Plus de cohésion entre le marketing produit et le développement => plus rapidement plus de qualité
- Plus de visibilité apportée au management (parfois de la « burning visibility »)

Merci pour votre attention !

Questions ?