COMP2511

Object-Oriented Design and Programming

Agile Software Processes

Wayne Wobcke

w.wobcke@unsw.edu.au

Today's Lecture

- Performance Management
- Toyota Production System
- Agile Manifesto
- Extreme Programming
- Scrum
- Project

Performance Management

- Why don't staff like it?
- Uncontrollable external factors
- Many tasks can't be measured
- Most outcomes result of many people
- Subjective, political, unfair, inequitable, rewards some things not others, fails to account for important efforts

Performance Management

- "But, as a manager, you cannot allow these impediments to keep you and your colleagues from working toward an effective incentive system" (HBS, 2006)

Toyota Production System

- “The Toyota Way is More than Tools and Techniques” (Liker, 2004)

Toyota Production System

- Lean manufacturing: Eliminate waste
  - Muda (waste) – 8 types
- Heijunka: Level out the workload
  - Muri (overburdening) – breakdowns, errors
  - Mura (unevenness) – irregular schedules
- Avoid overproduction
  - Kanban (sign; signboard)

Toyota Production System

- Build in quality
  - Jidoka (machine + human system)
  - Stop the line to fix problems
- Standardized work
  - Stable, repeatable methods, best practices
- Simple visual controls
  - Don’t hide problems
  - Technology supports people

Toyota Production System

- Management
  - Genchi genbutsu (actual place/thing)
  - Gemba (actual situation)
- Nemawashi
  - Make decisions slowly by consensus
  - Implement rapidly
- Organizational learning
  - Kaizen (continuous improvement)
  - Hansei (reflection)

Agile Manifesto

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan
- “That is, while there is value in the items on the right, we value the items on the left more” (www.agilemanifesto.org)

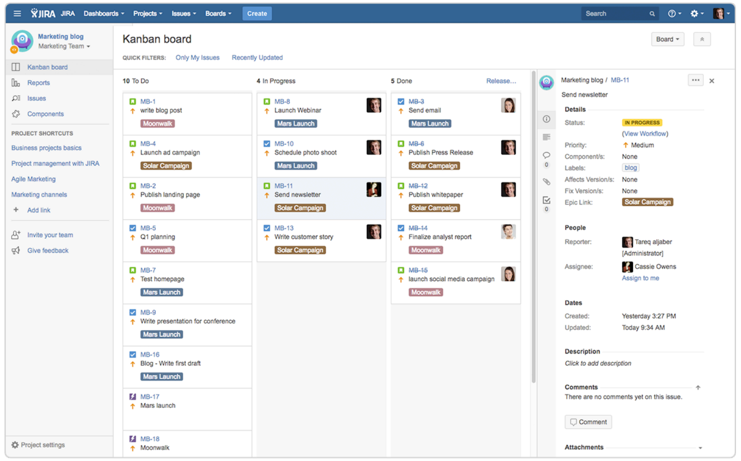
User Stories

- Short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system
- Typically follows a simple template
  - *As a <type of user>, I want <some goal> so that <some reason>*
- Often written on index cards or sticky notes, stored in a shoe box, and arranged on walls or tables to facilitate planning and discussion
- Strongly shifts the focus from writing about features to discussing them − the discussions are more important than whatever text is written

User Stories - Examples

- Epic (user story covering a large amount of functionality)
  - *As a user, I want to backup my entire hard drive so that I can restore my files in the case of a system failure*
- Broken down into smaller user stories
  - *As a power user, I want to specify files or folders for backup so that I can restore selected files*
  - *As a user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved*
- Conditions of Satisfaction/User Acceptance Tests
  - Conditions that must be fulfilled before the “customer” is happy

Kanban Task Board



Kanban board

10 To Do
- MB-1 write blog post — Moonwalk
- MB-4 Launch ad campaign — Solar Campaign
- MB-2 Publish landing page — Moonwalk
- MB-5 Q1 planning — Moonwalk
- MB-7 Test homepage — Mars Launch
- MB-9 Write presentation for conference — Mars Launch
- MB-16 Blog - Write first draft — Mars Launch
- MB-17 Mars launch
- MB-18 Moonwalk

4 In Progress
- MB-8 Launch Webinar — Mars Launch
- MB-10 Schedule photo shoot — Mars Launch
- MB-11 Send newsletter — Solar Campaign
- MB-13 Write customer story — Solar Campaign

5 Done
- ~~MB-3~~ Send email — Mars Launch
- ~~MB-6~~ Publish Press Release — Solar Campaign
- ~~MB-12~~ Publish whitepaper — Solar Campaign
- ~~MB-14~~ Finalize analyst report — Moonwalk
- ~~MB-15~~ launch social media campaign — Moonwalk

Marketing blog / MB-11
Send newsletter

Details

Status:	IN PROGRESS (View Workflow)
Priority:	Medium
Component/s:	None
Labels:	blog
Affects Version/s:	None
Fix Version/s:	None
Epic Link:	Solar Campaign

People

Reporter:	Tareq aljaber [Administrator]
Assignee:	Cassie Owens Assign to me

Dates

Created:	Yesterday 3:27 PM
Updated:	Today 9:34 AM

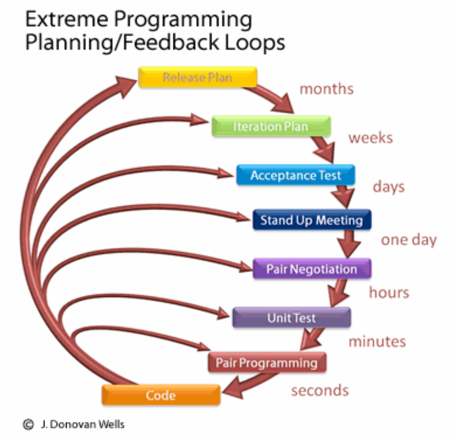
Description
Click to add description

Comments
There are no comments yet on this issue.
Comment

Attachments

Image http://www.atlassian.com/blog/marketing-teams/agile-marketing-fad-future-marketing

Extreme Programming



Extreme Programming Practices

Incremental planning	Requirements are recorded on story cards and the stories to be included in a release are determined by the time available and their relative priority. The developers break these stories into development 'Tasks'.
Small releases	The minimal useful set of functionality that provides business value is developed first. Releases of the system are frequent and incrementally add functionality to the first release.
Simple design	Enough design is carried out to meet the current requirements and no more.
Test-first development	An automated unit test framework is used to write tests for a new piece of functionality before that functionality itself is implemented.
Refactoring	All developers are expected to refactor the code continuously as soon as possible code improvements are found. This keeps the code simple and maintainable.

Extreme Programming Practices

Pair programming	Developers work in pairs, checking each other's work and providing the support to always do a good job.
Collective ownership	The pairs of developers work on all areas of the system, so that no islands of expertise develop and all the developers take responsibility for all of the code. Anyone can change anything.
Continuous integration	As soon as the work on a task is complete, it is integrated into the whole system. After any such integration, all the unit tests in the system must pass.
Sustainable pace	Large amounts of overtime are not considered acceptable as the net effect is often to reduce code quality and medium term productivity.
On-site customer	A representative of the end-user of the system (the customer) should be available full time for the use of the XP team. In an extreme programming process, the customer is a member of the development team and is responsible for bringing system requirements to the team for implementation.

Project

- Teams of **five** from same tut-lab – organized as a Scrum team
- Define, discuss, prioritize user stories – maintain on Trello
- Sprints are one or two weeks – sprint review in Week 10
- Design before coding – understandable UML class diagram
- Use git repository − everyone should contribute
- Show your diary to your tutor each week and e-mail at end
- Aim for minimum viable product first – extensions later
- Marking criteria for product
  - Correctness, Interaction, Design, Generality, Extensions
  - No provision for late submission – late submissions receive 0
  - Possibly adjusted using peer assessments entered into Moodle

Next Week

- User Centred Design & Usability
- GUI Programming