

Agile Project Inception

CSCI 5828: Foundations of Software Engineering
Lecture 10 — 02/16/2012

Goals

- Cover the material in chapters 3, 4, and 5 of our agile textbook
 - Agile Project Inception
 - How do you get an agile project started
 - What questions should you ask?
 - Who should be involved?
 - What tasks do you perform?

But first...

- Let's test our concurrency design skills
 - Imagine you have a list of Twitter user names (in the 100s)
 - You are asked to retrieve their most recent tweets (up to 3200 total)
 - And to store all of the returned tweets in a single file, one for each user
- Background
 - In order to retrieve the set of most recent tweets, you first ask for tweets associated with that user and get back 200 tweets
 - You then ask for “page 2”, “page 3”, up to “page 16”, for a total of 3200

Example

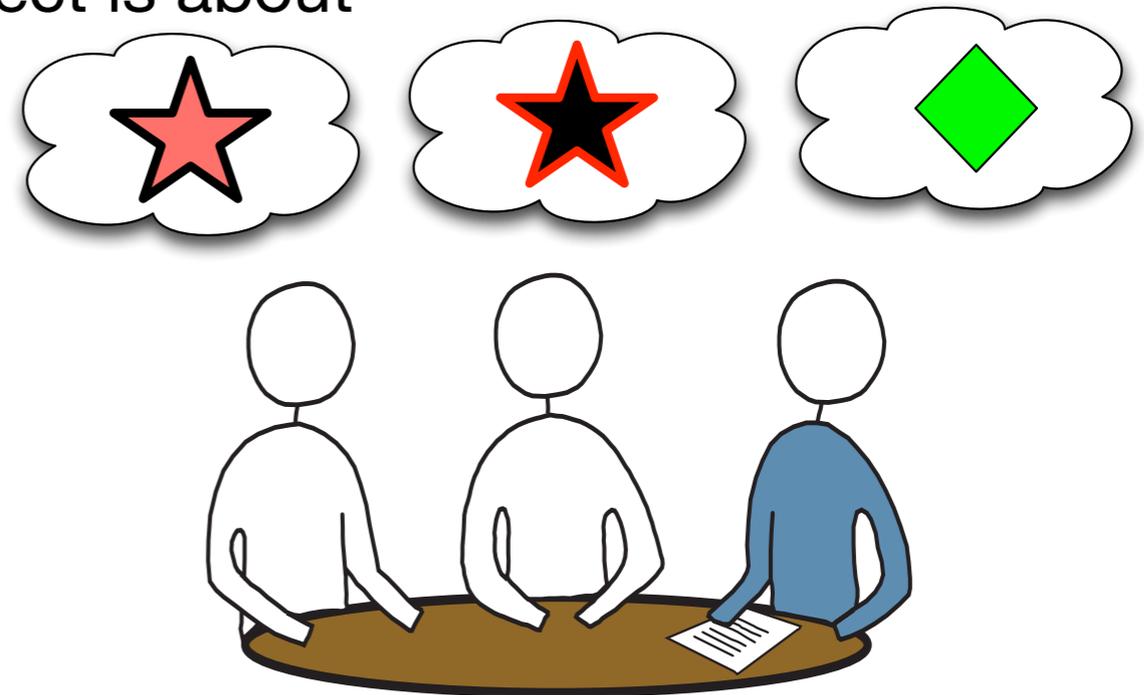
- To ask for the first “page” of 200 tweets from Markus Persson, a.k.a. notch
 - http://twitter.com/statuses/user_timeline/notch.atom?count=200&page=1
- This returns an XML document using the Atom format
- You can then change that URL to “&page=2”, “&page=3”, etc.
- Imagine you need to write a concurrent program to now do this for 100s of users, creating one file of 3200 tweets per user, on a 4 core machine
 - how would you design this concurrent system?
 - what tasks do you need?
 - what potential errors might you need to handle?

Getting a Project Started

- A key problem to getting a development project started is getting all stakeholders to understand what the project is about

- Challenges include

- What is the purpose of the system?
- What is its core functionality?
- What functionality is out-of-scope?
- What does success look like?



“We are in agreement, right?” “Sure!”

- If you fail to ask the right questions, and you fail to reach a shared understanding, then you’re setting the project up to fail
 - As our book says, “The assumption of consensus where none exists is what kills most projects”

Achieving consensus

- To achieve consensus, you need to gather information that
 - communicates the goals, purpose, and context of the proposed project to the team (stakeholders and developers)
 - so they can make intelligent decisions about whether the project is feasible and worthwhile
- In order to gather this information, you need to ask tough questions of your project's stakeholders (the start of a project is the best time to ask tough questions)
 - You then gather the results in the **inception deck**
 - A project initiation method used at ThoughtWorks, an IT consultancy company that provides Agile training and custom software development

Important Questions

- Setting aside the notion of technical feasibility, there are critical questions to ask before you commit to a new project
 - Can you assemble a team with the necessary skills?
 - How much money is available to fund the project?
 - Who's the customer?
 - Who's our competition?
- The inception deck provides additional questions to ask as well as activities to help induce shared consensus about what the proposed project is all about

Inception Deck (I)

- The inception deck is a collection of questions to ask of yourself and your stakeholders when booting up a project
 - The idea is to get the right group of stakeholders into a room and
 - have them perform a series of exercises
 - designed to elicit the expectations people have for the project
- Which stakeholders?
 - Anyone who can materially contribute to the execution of the project
 - customers (end users), developers, testers, managers, CFOs, CIOs, etc.
- The purpose: to provide enough information that a “go/no go” decision can be made; it will take anywhere from a few days to several weeks to do this

Inception Deck (II)

- The Inception Deck exercises produces a set of slides that can be hung up by the team on the walls of their shared workspace
 - The deck should be good for roughly six months of planning
- The deck is a living document; it should be revisited if any of the major elements of a project changes
- The ten questions discussed here are just the beginning
 - More questions will come up and need to be dealt with over the course of answering the primary ten questions

Inception Deck (III)

- At a high level, the inception deck will have your team

- ask “why are we here?” What’s the purpose of the project?

- create an elevator pitch

- design a product box

- create a NOT list

- meet their neighbors (project community)

- show the solution (UML-based architecture diagrams)

- ask what keeps them up at night? (What are the biggest risks?)

- size it up (how long do we think this project will last?)

- be clear on what’s going to give when time and/or money has run out?

- show what it’s going to take to be successful

the why

the how

Why are we here?

- The team needs to be able to identify why they are working on this system
 - If they understand the why, they are more likely to
 - make better decisions
 - balance trade-offs
 - pull together as a team and come up with more innovative solutions
- It's important to go and see the environment in which your system will be deployed; such visits can be seen as a form of early requirements gathering
 - just enough to establish the importance of the problem and the complexity of the problem domain
- In addition, we should be able to express our purpose concisely
 - Example from Southwest Airlines: Will this project lower the price of a ticket?

Create an Elevator Pitch

- Work on being able to express the major aspects of your system in a short amount of time
 - The basic template is
 - For [target customer]
 - who [needs X]
 - the [product name] is a [product category]
 - that [key benefit]
 - Unlike [primary competitor], our product [statement of differentiation]
- This exercise will force you to be short and concise and know who your customer is and how you're helping that person (or persons)

Design a Product Box

- Ask why someone would buy your product?
 - The answer should be meaningful enough that it would appear on the box that is helping to sell the system you want to produce
 - Identify the system's benefits (not its features)
 - Create a slogan for the system (how will you sell it?)
 - Design the actual box: what pictures would you use and what name would you use; how would you display the slogan and the benefits

Create a NOT List

- It is critical to establish the boundaries of your project
 - What is “in scope”?
 - What functions will be part of our system?
 - What is “out of scope”?
 - What functions will definitely not be considered?
 - Can you define this boundary succinctly such that there are no ambiguities?
 - What is “unresolved”?
 - What functions and/or requirements do we have that are currently ambiguous?
- Key point is this list is visual; up on a slide for all people to see

Meet Your Neighbors

- Reflect on who is a core member of the team
 - who will work on the project day in and day out
- And who is not a core member but is still an important stakeholder
 - This latter group will identify your project's community
 - Groups and/or individuals that you will need to interact with before your system goes live
 - Legal, database administrators, technical writers, external vendors, etc.
- The groups that end up on this list are entities that you will need to start developing professional relationships with
 - You will want them to support the project when it comes time for their concerns or expertise to play a role in completing the project

Show Your Solution

- Even though you still have requirements to gather and the topic domain to understand, most teams can still cobble together a diagram that
 - shows the boundaries and users of the system
 - shows the big components of the system
 - shows how these components are related
 - shows which components would be used by which type of user
- Such an exercise can help develop consensus about
 - the tools and techniques needed to implement the system
 - what is the scope of the system
 - what are the chief risks of the system

What Keeps You Up At Night?

- Identify the risks facing the project
 - What are the key challenges
 - with respect to technology, team skills, and stakeholders?
- Perhaps
 - your developers don't have the right skill set
 - your customer doesn't appear 100% committed to the project
 - there may be no clear technology choice for the constraints you are facing
- Talking about risk is good, as it helps to build consensus and awareness of the challenges facing the project right from the start

Size It Up

- Try to develop a consensus for how big of a project this is
 - Will it take 1 to 2 months? 3-4? 6 months?
 - If you think it will take more than 6 months, the book recommends that you reconsider the project
 - how can you make it smaller to fit within a 6 month timeline
 - Trying to plan out beyond six months for most projects doesn't make sense
 - too many things can change in that timeframe
- As part of this task, you will develop a rough cut schedule
 - how much time for development, user testing, training, etc.

Be Clear on What's Going to Give

- On projects, there are four forces that always come into play
 - **time** (what's our deadline?)
 - **budget** (how much money do we have?)
 - **quality** (what practices do we have in place to ensure good value for our customer?)
 - **scope** (how much are we trying to accomplish?)
- If you start to run out of time or money, you have to identify what is going to give on the project
 - assuming that time and money are fixed and that our commitment to our customer will not allow us to reduce the overall quality of the system
 - then the only thing left to “flex” is scope: what it is we are trying to do

Show What It's Going to Take (I)

- Having settled on the vision for the project, it's size, and a basic plan on how to proceed
 - the last thing you need is to state how you are going to achieve the goals presented so far
- You need to specify the team, the cost, and the overall schedule
 - You need to answer the questions
 - When is it going to be done? How much is it going to cost?
 - Your costs will be driven primarily by the team (salary) multiplied by the projected time the project will take
 - there will also be (relatively minor) start-up costs related to equipment and services

Show What It's Going to Take (II)

- With respect to the team, you need to outline the key roles and identify the people who will play them
 - You also need to identify which stakeholder will play the role of the customer
 - You then need to ensure that the person or persons identified as the customer understand the role they need to play
 - that they supply
 - the requirements (and their interpretation)
 - the priorities
 - and feedback on working software
- The project cannot succeed without them!

Summary

- Before we can start to gather requirements, we need to determine if a project is feasible
 - Since a lack of shared consensus about a project can kill it, we reviewed a technique used by the author of our agile textbook for helping to kickstart a project
- The inception deck provides ten questions to ask / activities to perform that help
 - identify the purpose and benefits of a project
 - who is involved and why
 - what are the risks, the proposed solution, and how much is it going to cost

Coming Up Next

- Lecture 11: Taming Shared Mutability, Part 1
- Lecture 12: Taming Shared Mutability, Part 2