Agile Software Development

Georgia Tech

Agile Processes

Agile Manifesto (http://agilemanifesto.org/) value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

Guiding principle: change happens, so organize your process around that fact

- Some developers say "we could build good software if only we had stable requirements"
- Agile developers recognize that requirements aren't stable and adjust their practices to that reality

Low ceremony, emphasis on working software over "software development bureaucracy"

Georgia Tech

eXtreme Programming

Agile method characterized by 12 practices:

- The Planning Game
- Small releases
- Metaphor
- Simple design
- Testing
- Refactoring
- Pair programming
- Collective ownership
- Continuous integration
- 40 hour week
- On-site customer
- Coding standards

Based on already well-known "best practices." XP takes them "to the extreme"

Georgia Tech

XP Practices

- The Planning Game – Quickly determine the scope of the next release by combining business priorities and technical estimates. As reality overtakes the plan, update the plan.
- Small releases – Put a simple system into production quickly, then release new versions on a very short cycle.
- Metaphor – Guide all development with a simple shared story of how the whole system works.
- Simple design – The system should be designed as simply as possible at any given moment. Extra complexity is removed as soon as it is discovered.
- Testing – Programmers continually write unit tests, which must run flawlessly for development to continue. Customers write tests demonstrating that features are finished.
- Refactoring – Programmers restructure the system without changing its behavior to remove duplication, improve communication, simplify, or add flexibility.

[^1] From Extreme Programming Explained, by Kent Beck

Georgia Tech

XP Practices (2 of 2)

- Pair programming – All production code is written with two programmers at one machine.
- Collective ownership – Anyone can change any code anywhere in the system at any time.
- Continuous integration – Integrate and build the system many times a day, every time a task is completed.
- 40 hour week – Work no more than 40 hours a week as a rule. Never work overtime a second week in a row.
- On-site customer – Include a real, live user on the team, available full-time to answer questions.
- Coding standards – Programmers write all code in accordance with rules emphasizing communication through the code.

Georgia Tech

Scrum

Scrum is a framework for organizing and managing work.

- Focuses on management of software development process rather than practices
- Timeboxed iterations called sprints
- Timeboxed means the schedule is firm. If something doesn't get done, it goes into the next sprint.
- Daily Meeting (Pigs and Chickens)
- Each sprint ends with a release, a "ready" deliverable that includes a subset of the final product features

Definition of "Done": team and project dependent. Typically, when customer accepts a story (usually also means the functional tests pass).

For a nice concise guide to Scrum, see The Scrum Guide. We'll summarize in the next few slides.

Georgia Tech

The Scrum Team

- **Product Owner** is the sole person responsible for managing the Product Backlog. May delegate to deve team, but still responsible.
- **Development Team** consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint. Only members of the Development Team create the Increment.
- **Scrum Master** is responsible for ensuring Scrum is understood and enacted.

Georgia Tech

Scrum Events

- Planning: product vision and product backlog in the form of epic user stories.
- Sprints: building the product in iterations.

The heart of Scrum is the Sprint, a time-box of one month or less during which a "Done", useable, and potentially releasable product Increment is created. Each sprint has a goal, or a single coherent theme for the sprint that focuses the team.

Georgia Tech

Envisioning, a.k.a. Ideation, a.k.a. Product Planning

Participants

- Required: customer
- Recommended: Scrum team
- Vision will be refined in sprints with help of Scrum team

Inputs

- Initial idea or pivoted idea
- Planning horizon
- Completion date
- Budget/resources
- Confidence threshold – definition of "done", when you have enough for management to make a go/no-go funding decision

Outputs

- Product vision
- Epic stories (would be in icebox in Pivotal Tracker)
- Product roadmap (optional)

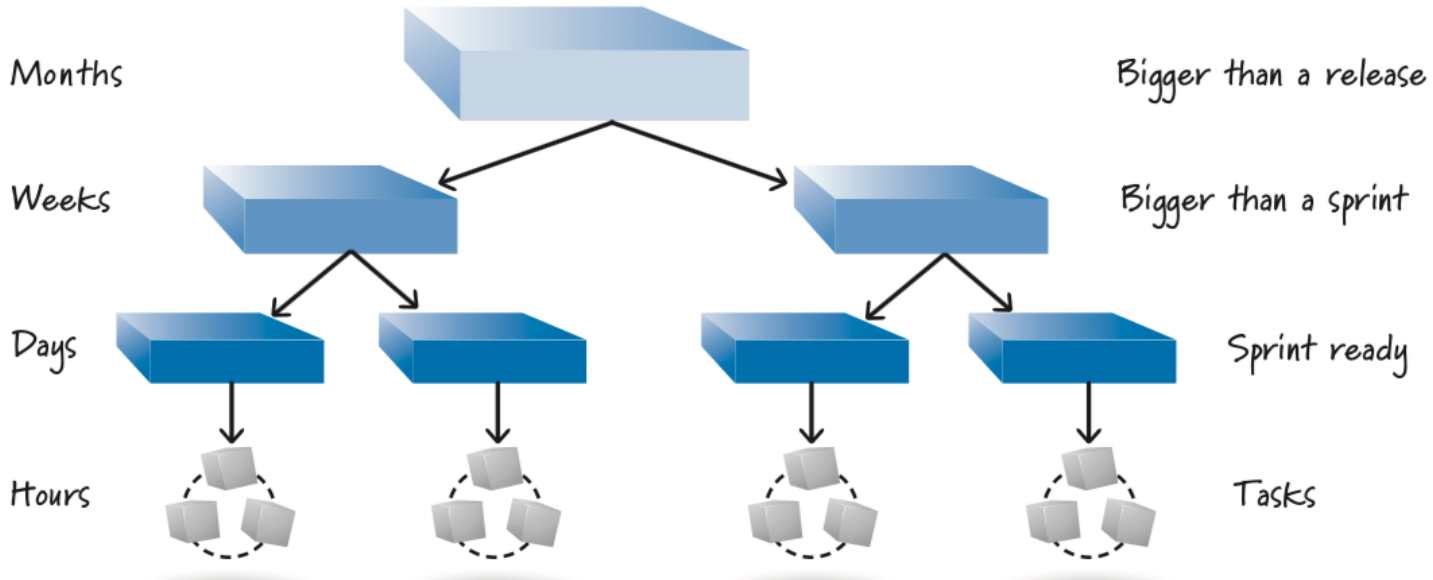
Georgia Tech

Product Vision Formats

- **Elevator statement** Write a 30-second to one-minute quick pitch of the product vision. Imagine you have stepped into an elevator with a venture capitalist and you have to pitch him on your product vision. Could you do it in a short elevator ride?
- **Product datasheet** Write the product datasheet on the first day. Try to fit it on the front side of a one-page marketing piece.
- **Product vision box** Draw the box in which you want to put the product when it ships. Can you come up with three or four salient points to illustrate on the box? (Drafting 15 points is easier than drafting three or four.)
- **User conference slides** Create the two or three presentation slides that you would use to introduce the product at your user conference (or equivalent). Try to avoid any bullet points on your slides.
- **Press release** Write the press release you want to issue when the product becomes available. Good press releases clearly communicate what is newsworthy in one page or less.

Epic Stories

A large user story, perhaps a few to many months in size, that can span an entire release or multiple releases. Epics are useful as placeholders for large requirements. Think: features.



Epics must be broken down into "sprintable" stories that go into Sprint Backlogs. We'll cover these stories when we discuss Pivotal Tracker.

Georgia Tech

User Stories

\begin{quote} As a ***, I want to*** , so that I can ____ \end{quote}

Specifies

- type of user,
- the task they are trying to do, and
- why they want to do it.

In the description section, we put the actual interaction between the user and the system. Remember this is WHAT and not HOW.

Georgia Tech

Story Example

Epic: "Buy a Product"

User Stories:

- As a customer I want to add a product to my shopping cart so that I can continue shopping and come back to my cart later to check out.
- As a cutomer I want to edit the items in my cart so that I can change my mind without creating a new shopping session.
- As a customer I want to check out with my shopping cart so that I can pay for my products and have them shipped to me.
- ...

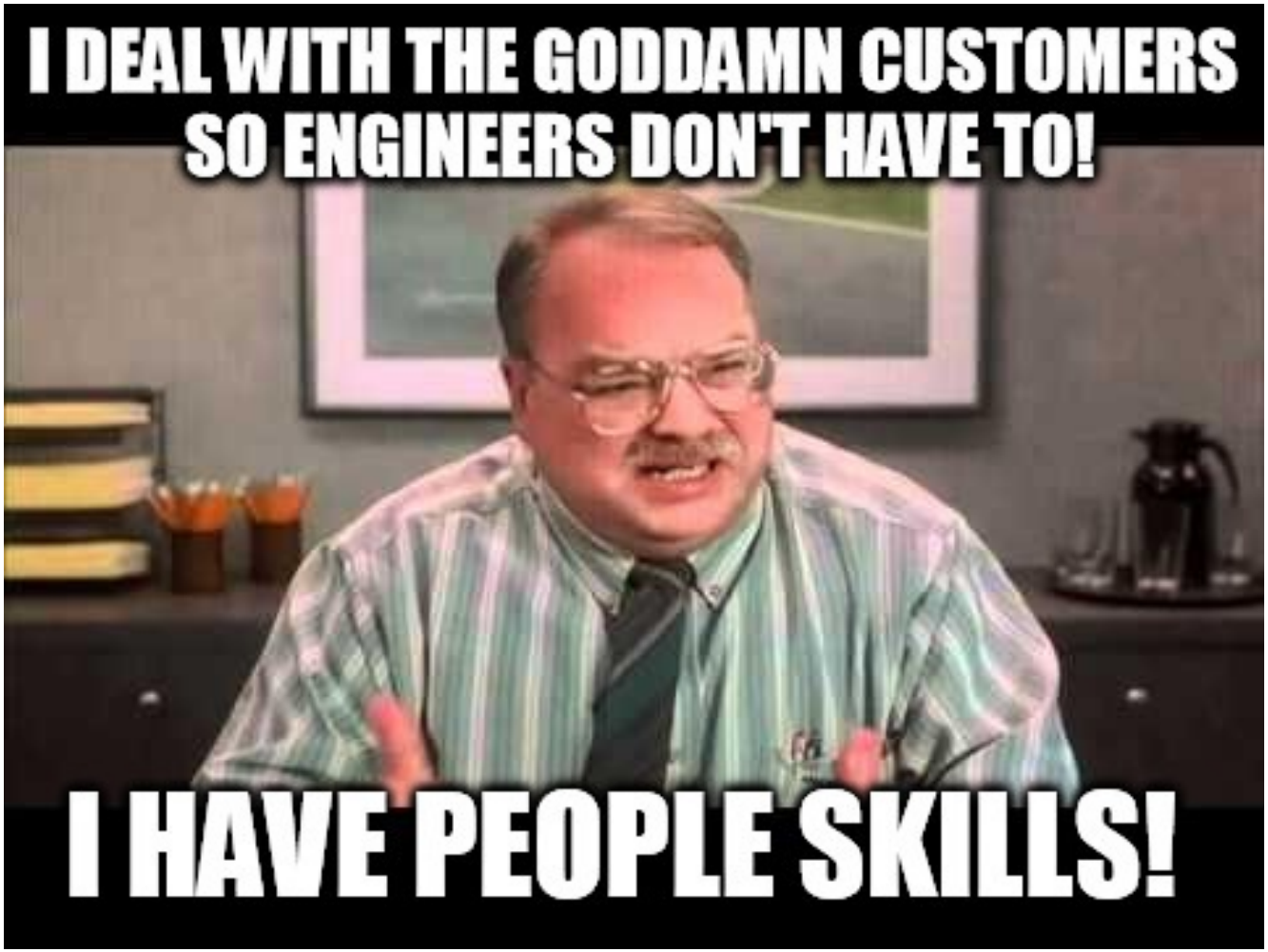
Georgia Tech

Writing Good Stories

- **I**ndependent – loosely coupled with other stories
- **N**egotiable – user stories placeholders for conversations with the customer, not a contract
- **V**aluable – value to the *customer* relative to other stories, permitting prioritization
- **E**stimatable –
- **S**mall (sized appropriately)
- **T**estable

Georgia Tech

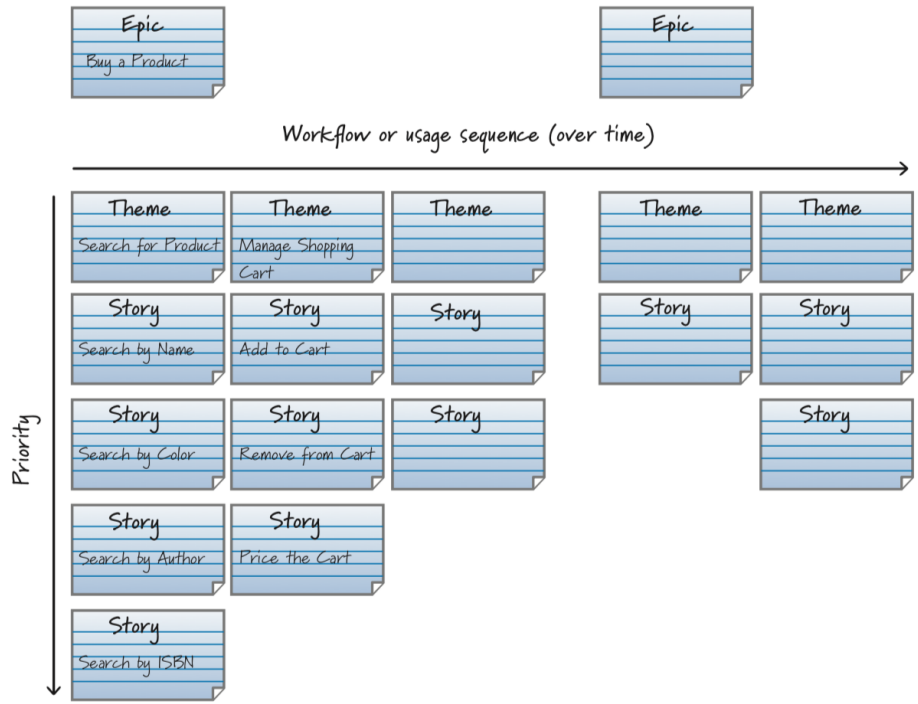
Gathering Stories



Georgia Tech

Gathering Stories

- The goal of a **user-story-writing workshop** is to collectively brainstorm desired business value and create user story placeholders for what the product or service is supposed to do.
- **Story mapping** takes a user-centric perspective for generating a set of user stories. The basic idea is to decompose high-level user activity into a workflow that can be further decomposed into a set of detailed tasks.



Sprint Events

- **Sprint Planning**: which stories will be implemented in this sprint, and how will they be implemented (design, tasks, etc.). Decompose epic stories into "sprintable" stories as needed.
- **Daily Scrum**: The Daily Scrum is a 15-minute time-boxed event for the Development Team to synchronize activities and create a plan for the next 24 hours. Every team member answers:
  - What did I do yesterday to help meet the Sprint Goal?
  - What will I do today to help meet the Sprint Goal?
  - Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?
- **Sprint Review**: held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed.
- **Sprint Retrospective**: assess the effectiveness of the Scrum team and identify areas for improvement.

Georgia Tech

Scrum Artifacts

- **Product Backlog**: an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.
- **Sprint Backlog**: the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal
- **Increment**: the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints.

Georgia Tech

Domain-Driven Design

We'll discuss DDD in more detail later, but you should start thinking about the basic tenets of DDD now.

- Focus on representing problem domain in code
- Bounded contexts
- Ubiquitous language

Georgia Tech

Implementing Scrum with Pivotal Tracker

An automated tool to help manage agile projects.

Divides work into:

- Chores: work that has to be done, but that doesn't deliver direct benefit to the customer
- Stories: pieces of functionality that form part of the actual application
- Bugs: malfunctioning parts of the system that need to be fixed.

Georgia Tech

Setting up Pivotal Tracker

- Go to http://www.pivotaltracker.com
- Create a public project (only public projects are free!)
- Add your team mates to the project
- Add me to the project (chris.simpkins@gatech.edu)
- Add your TA to the project once you know them

Georgia Tech

Creating Stories and Chores

- Press Add Story button
- Fill out information
- Automatically goes into the icebox
- Once a chore/story/bug has an estimate, it moves to the backlog
- Prioritize the backlog by dragging and dropping
- Most complex (largest estimates) stories go first.
- Prioritizing complex tasks is a risk management strategy – you have time to adapt

A User Story should take no more than half an iteration to complete. If a story takes an entire iteration, there is greater risk of not completing the story.

Georgia Tech

Estimation

Once a story is created it goes in the icebox, where we can estimate its required effort on an integral scale from 0–4. These are called story points.

A story point is an abstract measure of effort:

- For example a point might be an hour
- It might be a person-week
- It might mean "easy"
  - 1 = EASY
  - 2 = MEDIUM
  - 3 = HARD

Velocity is the number of story points you can accomplish in a single sprint. You guess this initially, then pivotal tracker calculates based on historical data after you've completed some sprints.

Georgia Tech

Story Life Cycle Categories

- **Icebox**, where every story starts. Once a story is estimated, it can be moved into the backlog.
- **Backlog**: the actual things you are going to do. We drag and drop things from the icebox into the backlog. The backlog should be in priority order, things at the top will be worked on before things on the bottom. You change priorities with simple drag and drop.
- **Current**: the highest priority things will be moved from the backlog to Current based on your velocity. You do not directly control what is put into Current.
- **Done**: accepted stories, chores and bugs are put into done. Again you don't directly move anything into done, it is automatic: developer team finshes and delivers, cusomter accepts.

Visually, stories move from right to left as they are ideated, estimated, implemented, completed, and accepted

Georgia Tech

A Day in a Team's Life

- Log into Pivotal Tacker.
- Pick the highest priority item in current that is not being worked on.
- Click Start button.
- Open story and add any tasks.
- If not already present, write out the test procedure.
- Implement the story.
- Click Finish when done.
- Click Deliver when all tests pass.
- Customer clicks accept or reject.

Georgia Tech

This Semester's Project

- Description
- Requirements
- Deliverables
  - Milestones
  - Peer evaluations

Georgia Tech