Lecture 4: Intro To Process

Michael Hilton

Administrivia

- Homework 1 due today: 11:49
- Homework 2 released

Homework 1 Issues – Revenge of the Caching

- Two commands to run Mayan container:
  - Command 1:
    - >docker run -d --name mayan-edms
    - --restart=always -p 80:8000
    - -v /docker-volumes/**mayan**:/var/lib/mayan mayanedms/mayanedms:**latest**
  - Command 2:
    - >make docker build //builds mayan edms 3.2.7
    - > docker run -d \ --name mayan-edms \ --restart=always \ -p 80:8000 \ [CONNECTION STUFF] -v /docker-volumes/mayan-edms/media:/var/lib/mayan \ mayanedms/mayanedms:**<version>**

Learning Goals

- Recognize the Importance of process
- Understand the difficulty of measuring progress
- Identify what why software development has project characteristics
- Use milestones for planning and progress measurement
- Understand backlogs and user stories

PROCESS

How to develop software?

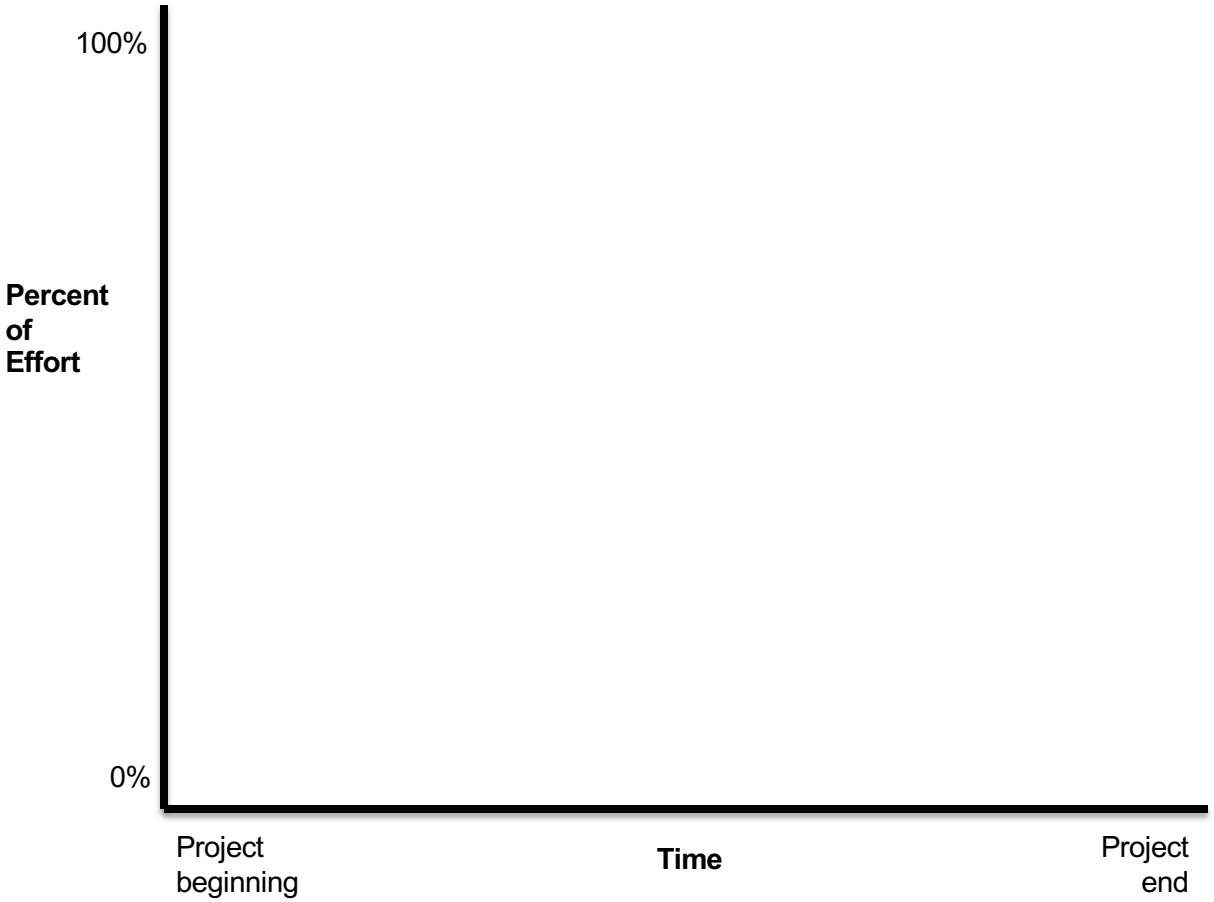
1. Discuss the software that needs to be written
2. Write some code
3. Test the code to identify the defects
4. Debug to find causes of defects
5. Fix the defects
6. If not done, return to step 1

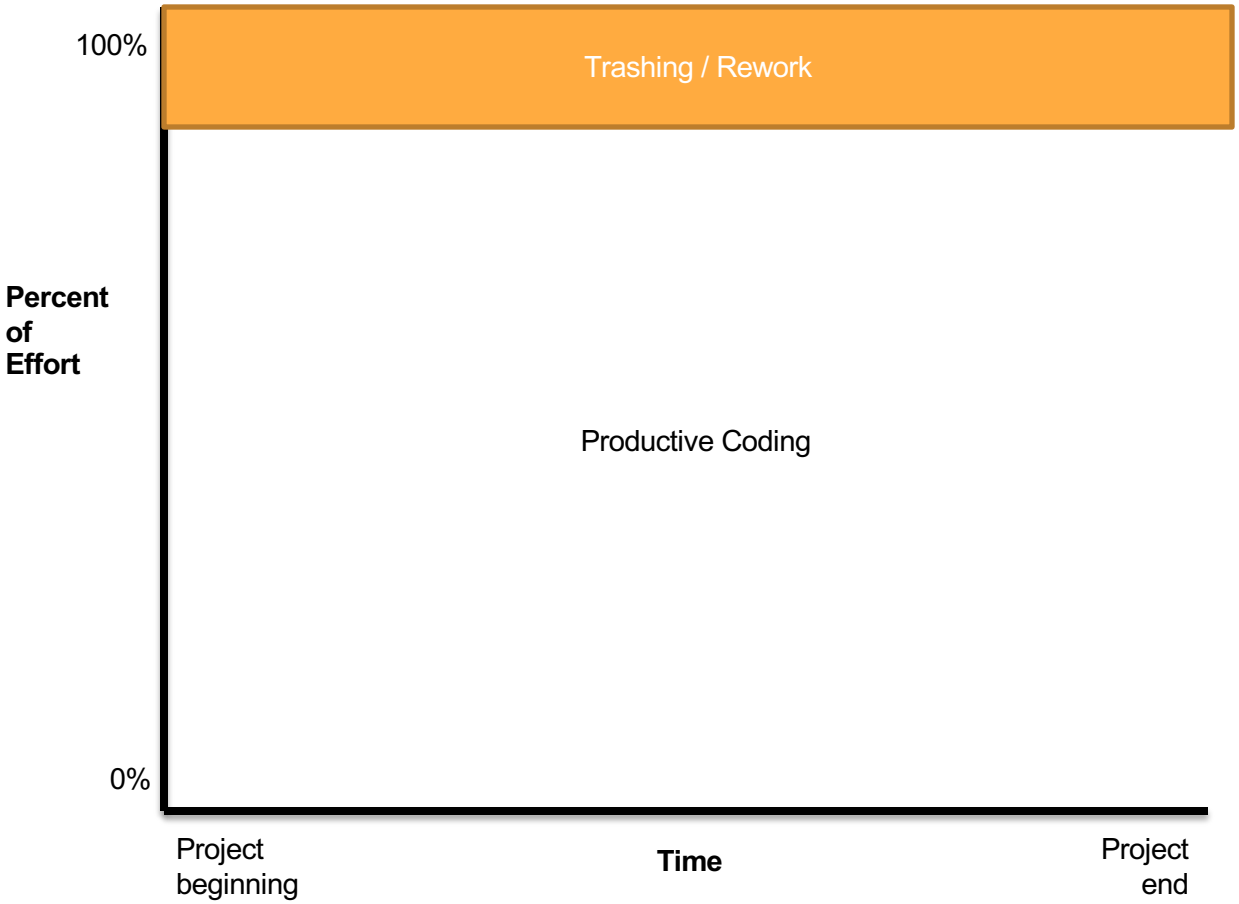
What is a Software Process?

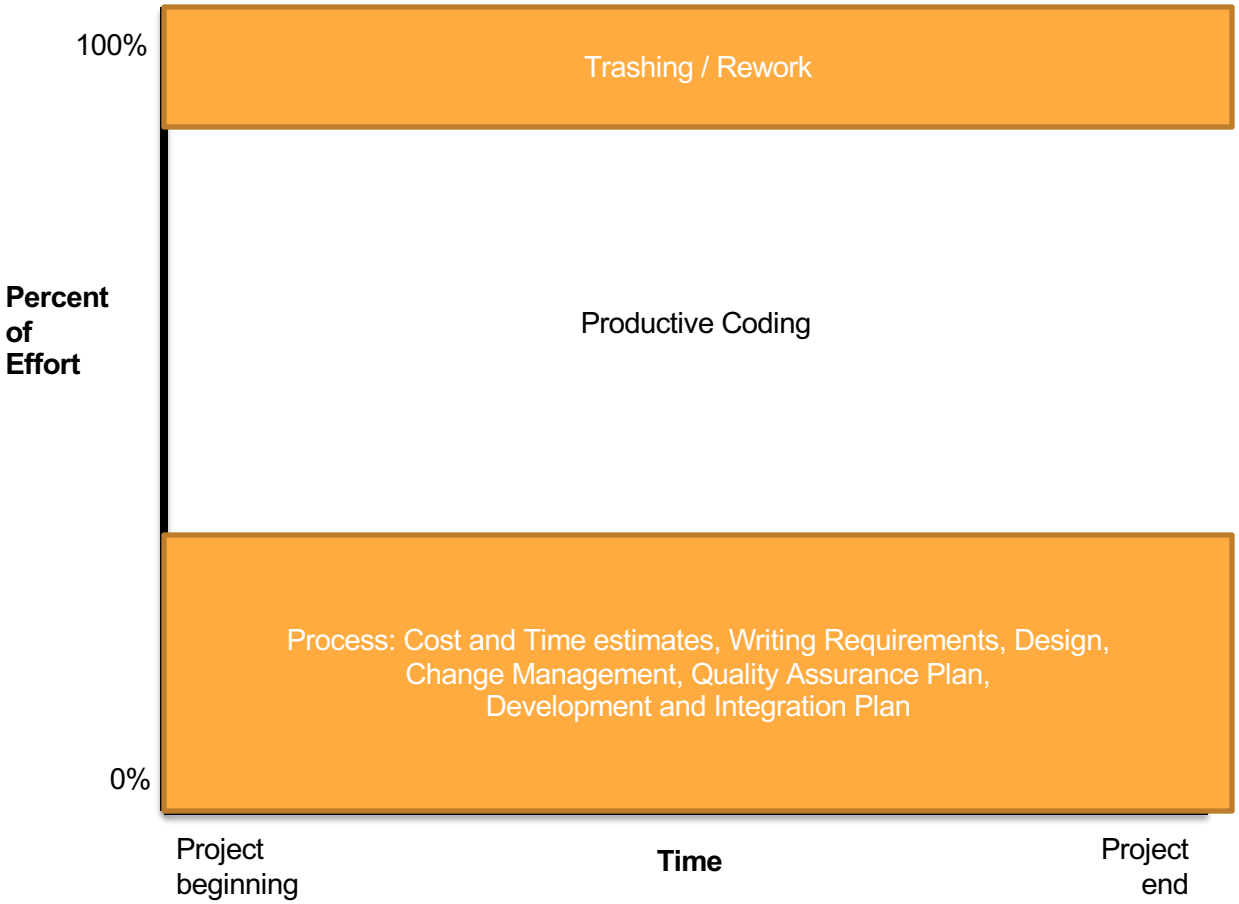
Software Process

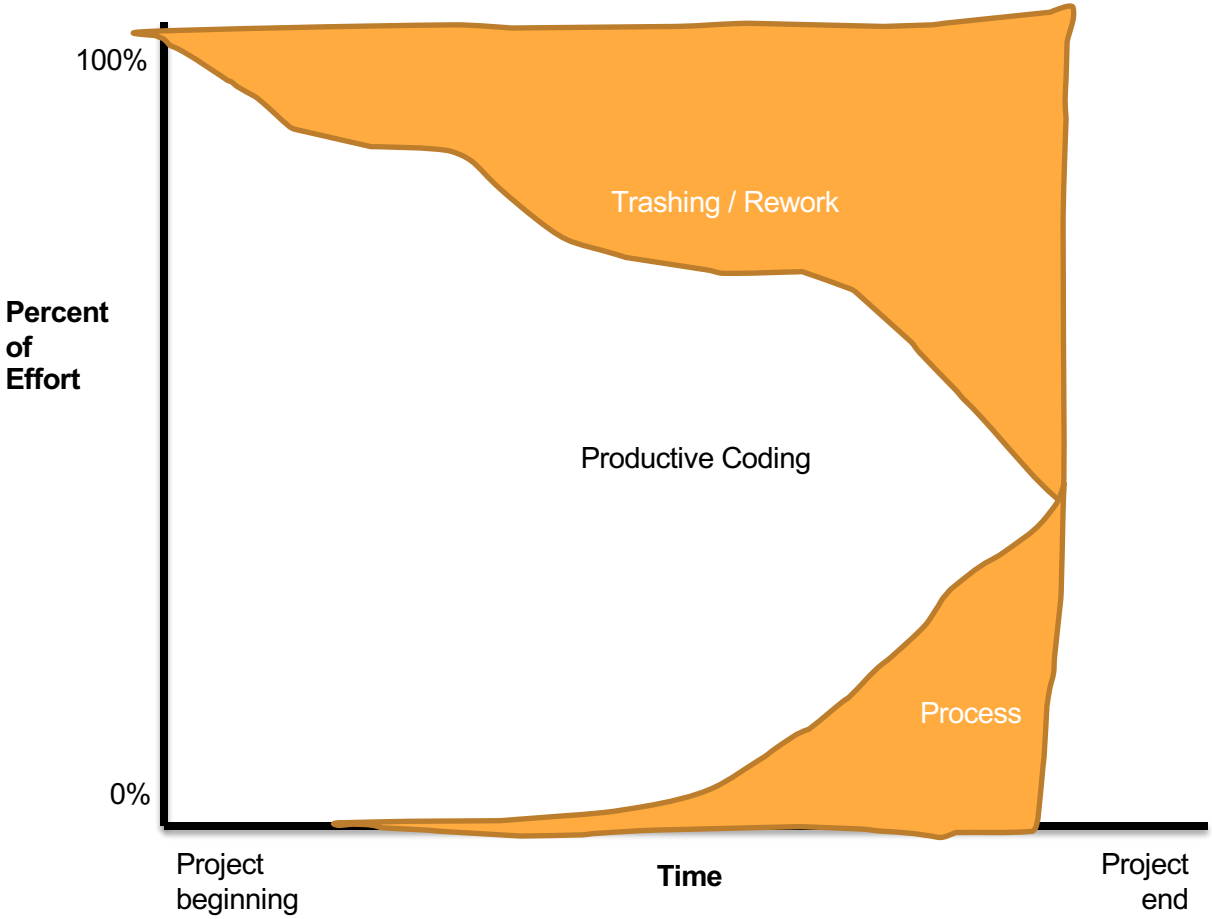
"The set of activities and associated results that produce a software product"

Sommerville, SE, ed. 8


100%
Percent of Effort
0%
Project beginning
Time
Project end


100%
Trashing / Rework
Percent of Effort
Productive Coding
0%
Project beginning
Time
Project end


100%
Trashing / Rework
Percent of Effort
Productive Coding
Process: Cost and Time estimates, Writing Requirements, Design, Change Management, Quality Assurance Plan, Development and Integration Plan
0%
Project beginning
Time
Project end

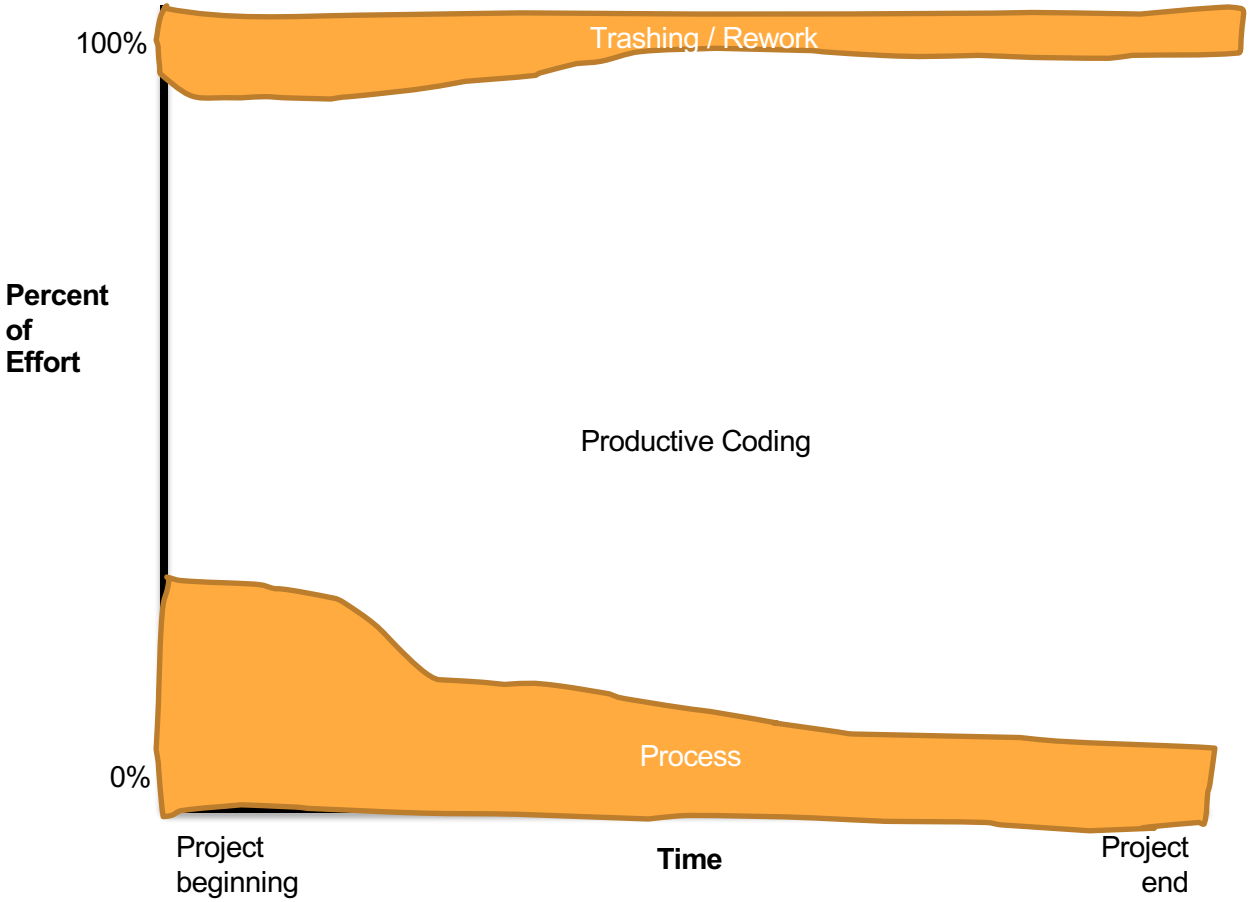

100%
Percent of Effort
0%
Trashing / Rework
Productive Coding
Process
Project beginning
Time
Project end

Example process issues

- Change Control: Mid-project informal agreement to changes suggested by customer or manager. Project scope expands 25-50%
- Quality Assurance: Late detection of requirements and design issues. Test-debug-reimplement cycle limits development of new features. Release with known defects.
- Defect Tracking: Bug reports collected informally, forgotten
- System Integration: Integration of independently developed components at the very end of the project. Interfaces out of sync.
- Source Code Control: Accidentally overwritten changes, lost work.
- Scheduling: When project is behind, developers are asked weekly for new estimates.

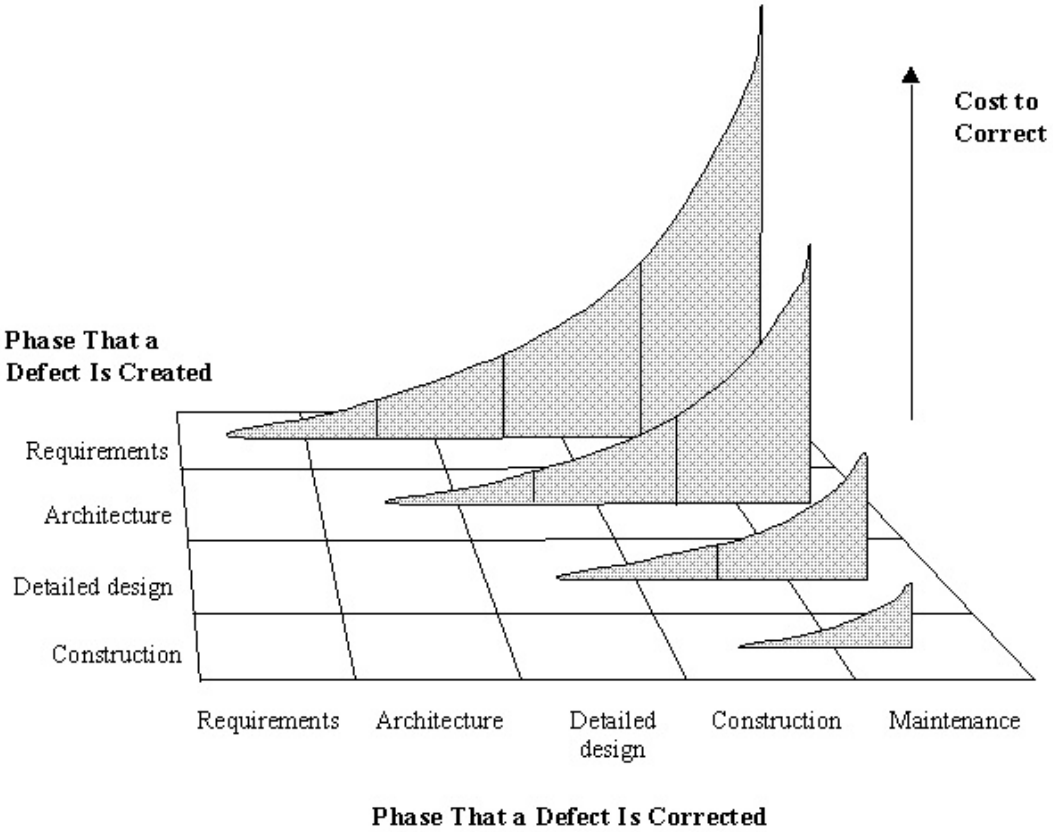
Survival Mode

- Missed deadlines -> "solo development mode" to meet own deadlines
- Ignore integration work
- Stop interacting with testers, technical writers, managers, …


100%
Trashing / Rework
Percent
of
Effort
Productive Coding
Process
0%
Project
beginning
Time
Project
end

Hypothesis

- Process increases flexibility and efficiency
- Upfront investment for later greater returns



Copyright 1998 Steven C. McConnell. Reprinted with permission from *Software Project Survival Guide* (Microsoft Press, 1998).

Planning

Estimating Effort

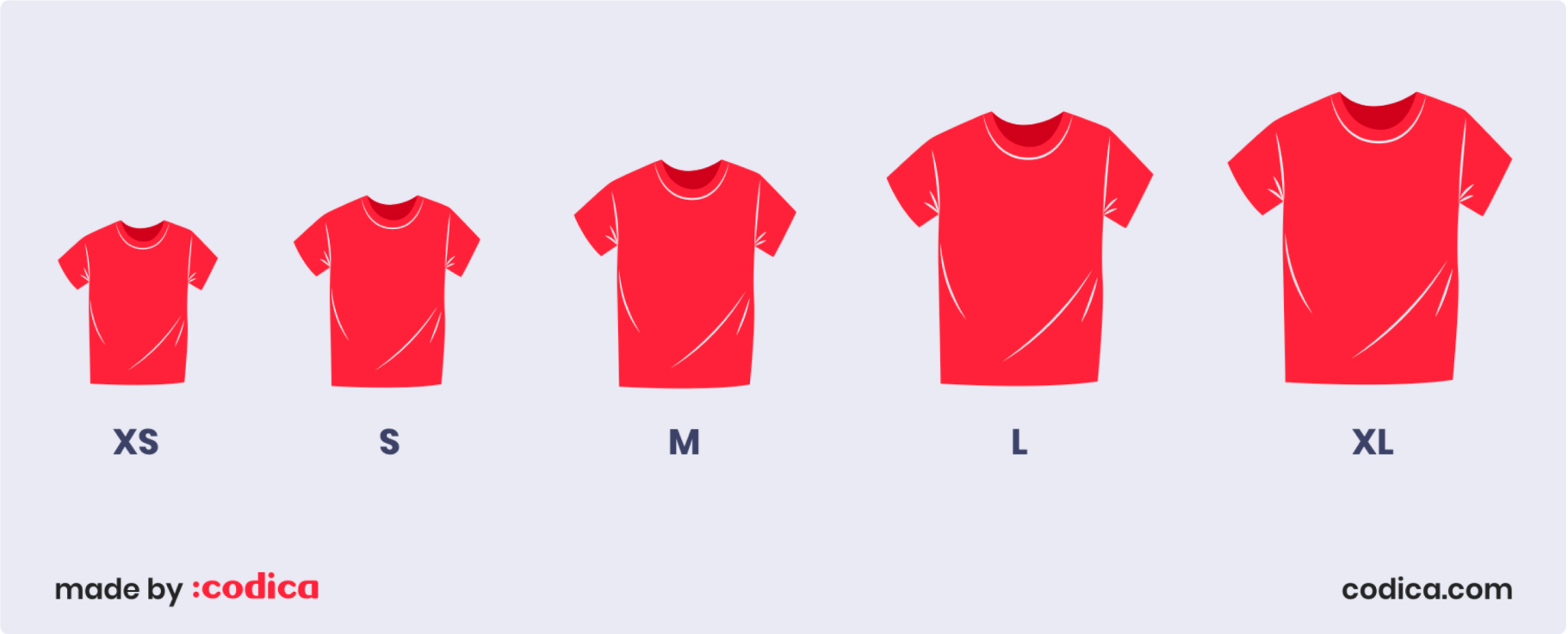
Task: Estimate Time

- A: Simple web version of the Monopoly boardgame with Pittsburgh street names
  - Team: just you
- B: Bank smartphone app
  - Team: you with team of 4 developers, one experienced with iPhone apps, one with background in security

- Estimate in 8h days (20 work days in a month, 220 per year)

Revise Time Estimate

- Remember Scrabble/Carcasonne experience?
- Is Monopoly similar/different/easier/more challenging/reusable?
- How much design did you do?
- Break down the task into ~5 smaller tasks and estimate them.
- Revise your overall estimate if necessary


XS
S
M
L
XL
made by :codica
codica.com

Measuring Progress?

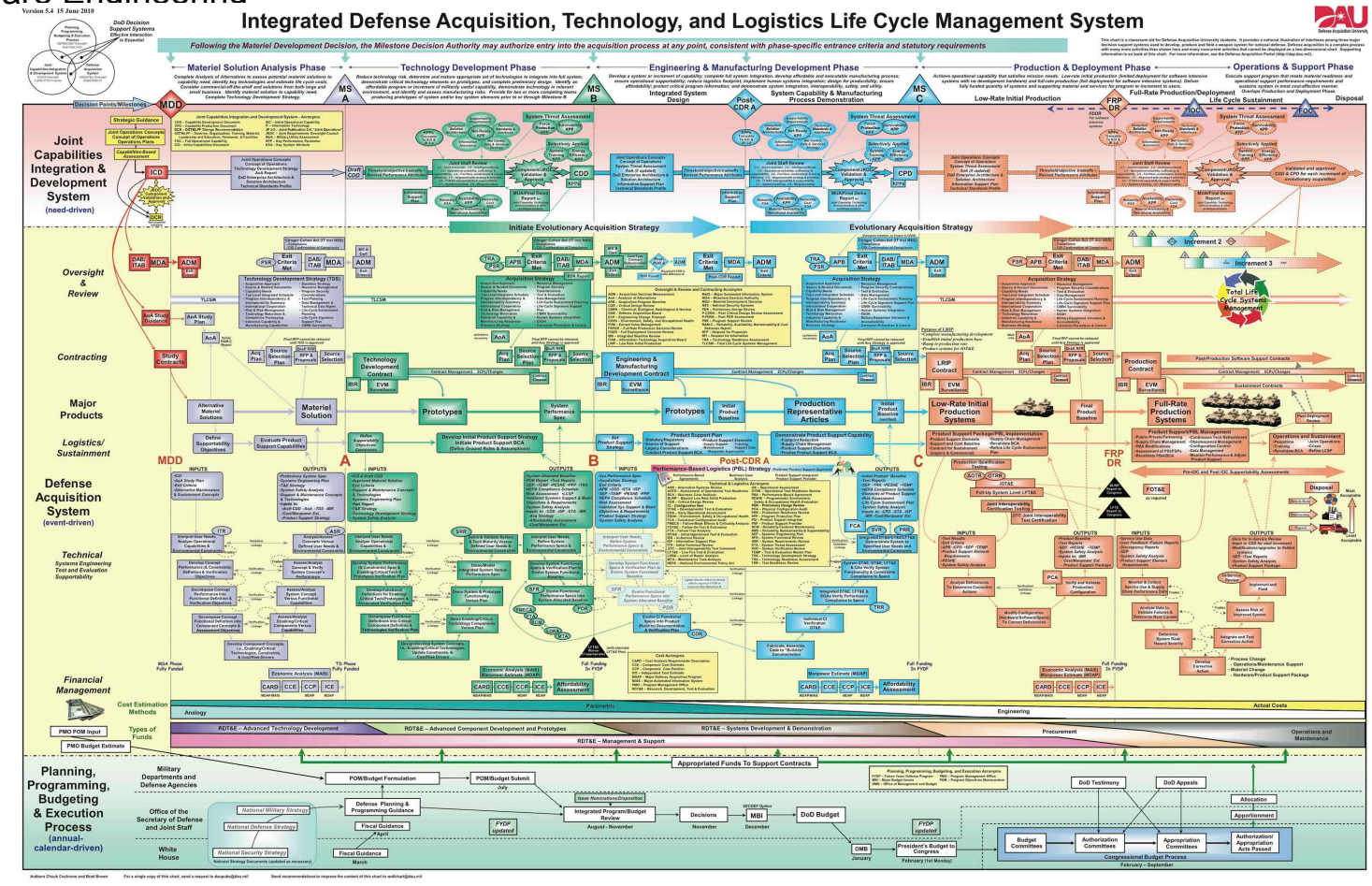
- "I'm almost done with the app. The frontend is almost fully implemented. The backend is fully finished except for the one stupid bug that keeps crashing the server. I only need to find the one stupid bug, but that can probably be done in an afternoon. We should be ready to release next week."

Measuring Progress?

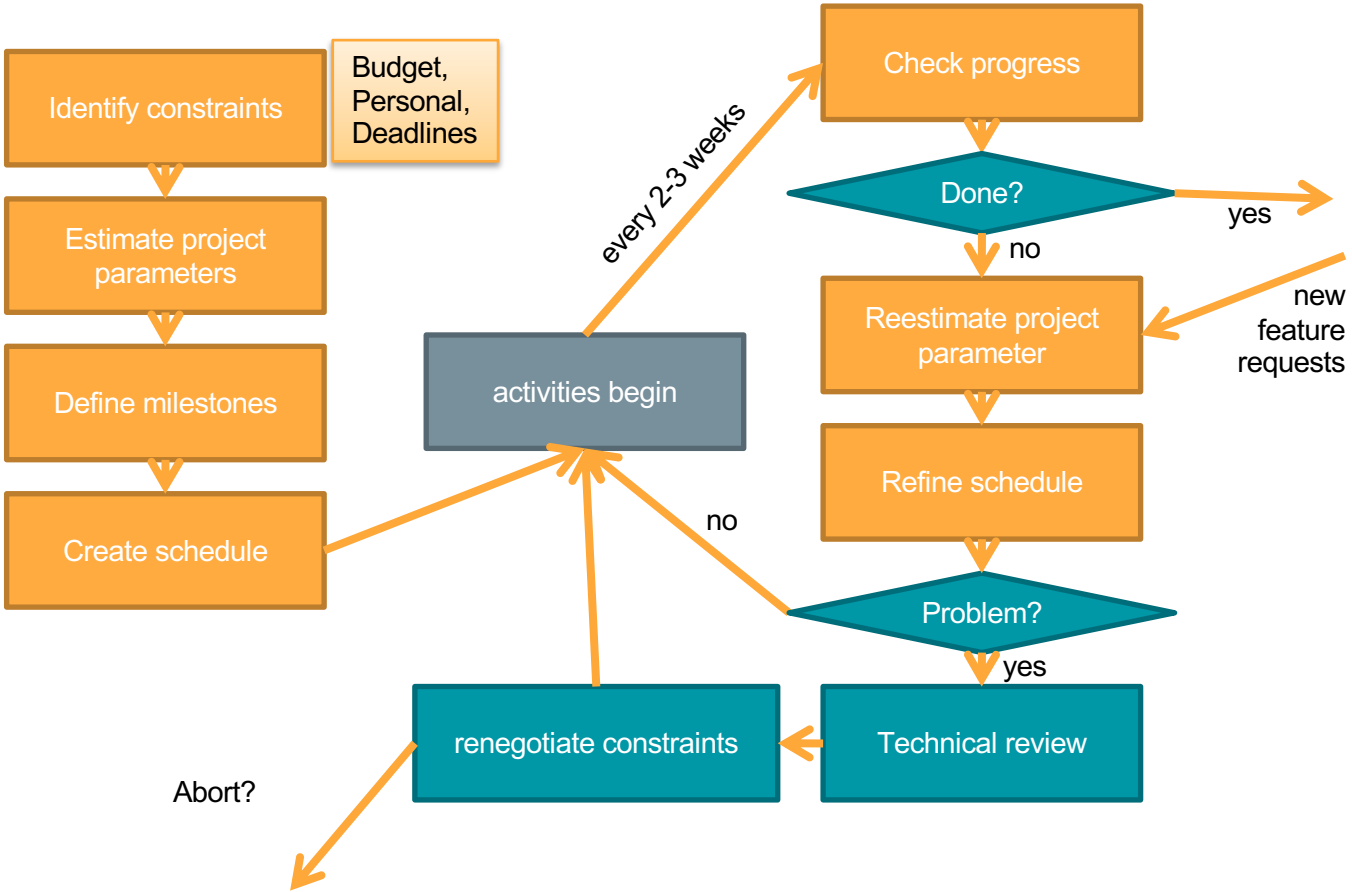
- Developer judgment: x% done
- Lines of code?
- Functionality?
- Quality?

Milestones and deliverables

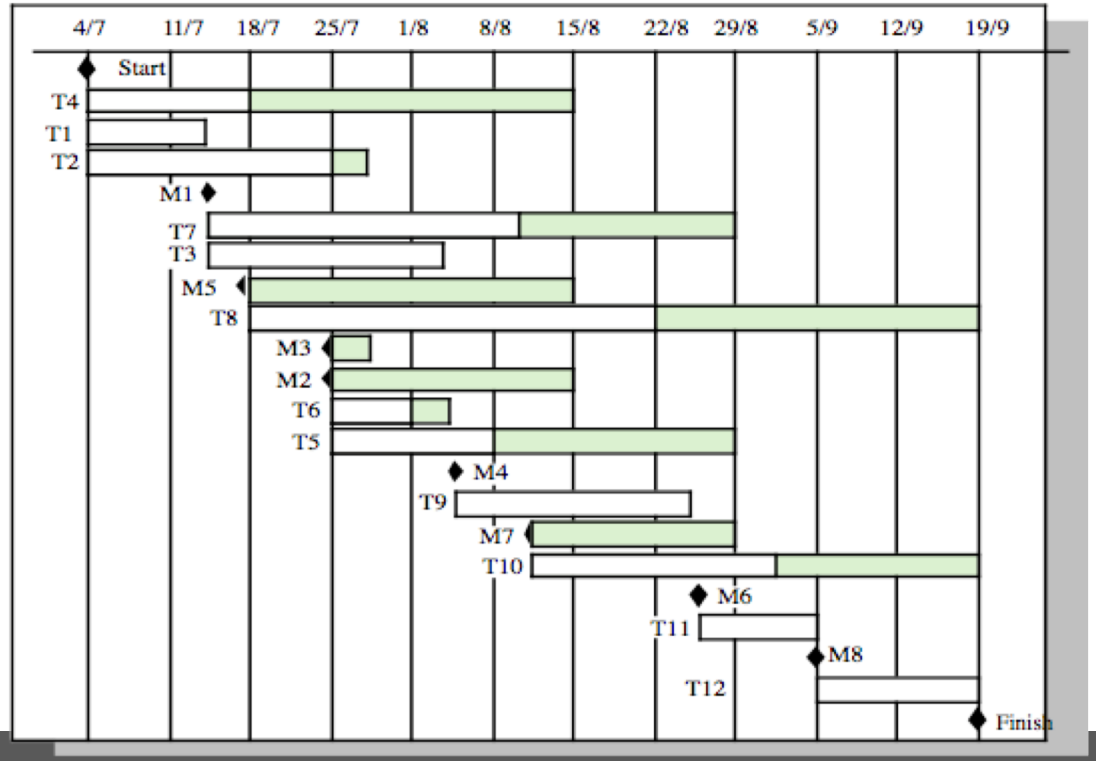
- Making progress observable, especially for software
- Milestone: clear end point of a (sub)tasks
  - For project manager
  - Reports, prototypes, completed subprojects
  - "80% done" not a suitable mile stone
- Deliverable: Result for customer
  - Similar to mile stone, but for customers
  - Reports, prototypes, completed subsystems


Integrated Defense Acquisition, Technology, and Logistics Life Cycle Management System
Following the Materiel Development Decision, the Milestone Decision Authority may authorize entry into the acquisition process at any point, consistent with phase-specific entrance criteria and statutory requirements
Materiel Solution Analysis Phase
Technology Development Phase
Engineering & Manufacturing Development Phase
Production & Deployment Phase
Operations & Support Phase
MS A
MS B
MS C
MDD
Post-CDR A
FRP DR
Full-Rate Production/Deployment
Low-Rate Initial Production
Disposal
Joint Capabilities Integration & Development System (JCIDS) (need-driven)
Oversight & Review
Contracting
Major Products
Logistics/ Sustainment
Defense Acquisition System (event-driven)
Technical
Systems Engineering Test and Evaluation Supportability
Financial Management
Cost Estimation Methods
Types of Funds
Planning, Programming, Budgeting & Execution Process (annual-calendar-driven)
Initiate Evolutionary Acquisition Strategy
Evolutionary Acquisition Strategy
Increment 2
Increment 3
Total Life Cycle Systems Management
Appropriated Funds To Support Contracts
Military Departments and Defense Agencies
Office of the Secretary of Defense and Joint Staff
White House
POM/Budget Formulation
POM/Budget Submit
Defense Planning & Programming Guidance
Integrated Program/Budget Review
Decisions
DoD Budget
President's Budget to Congress
Budget Committees
Authorization Committees
Appropriation Committees
Authorization/ Appropriation Acts Passed
Compressional Budget Process
DoD Testimony
DoD Appeals
Allocation
Apportionment

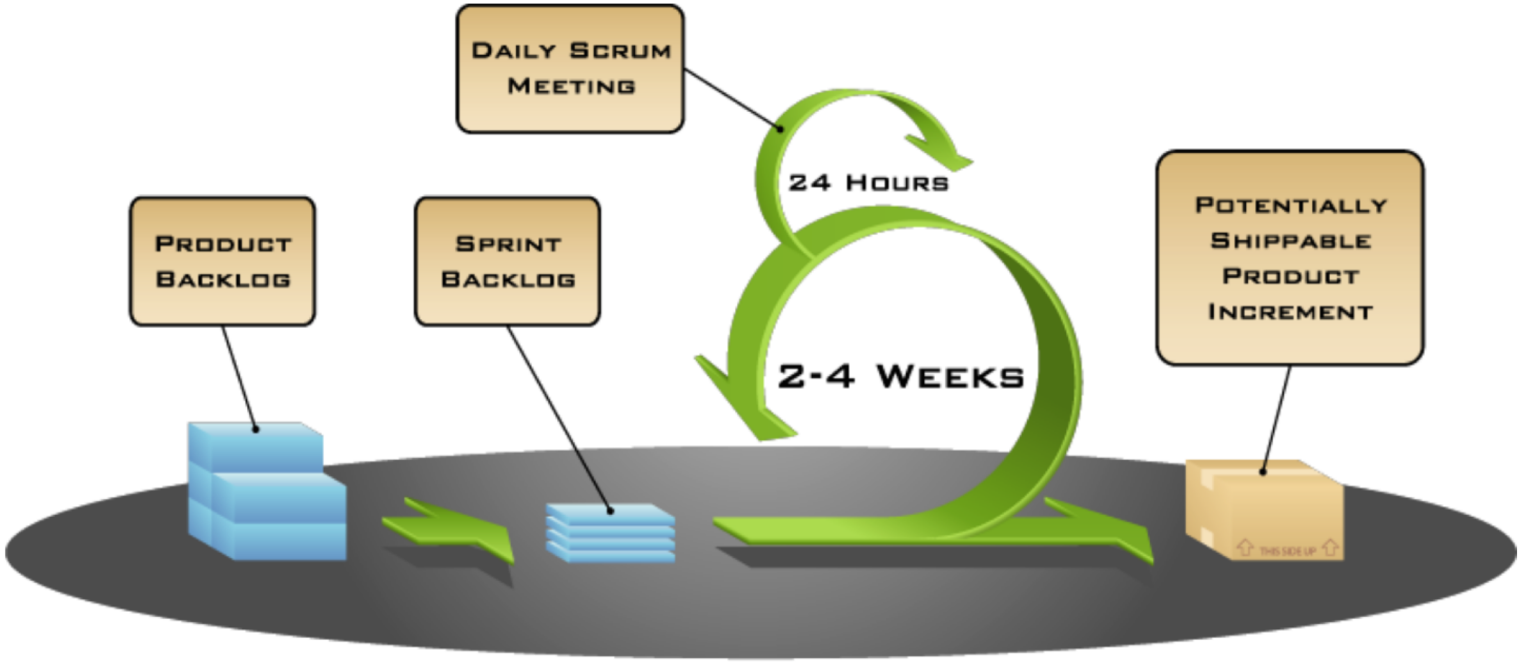
Project Planning



Gantt Diagrams



Brief intro to Scrum



Elements of Scrum

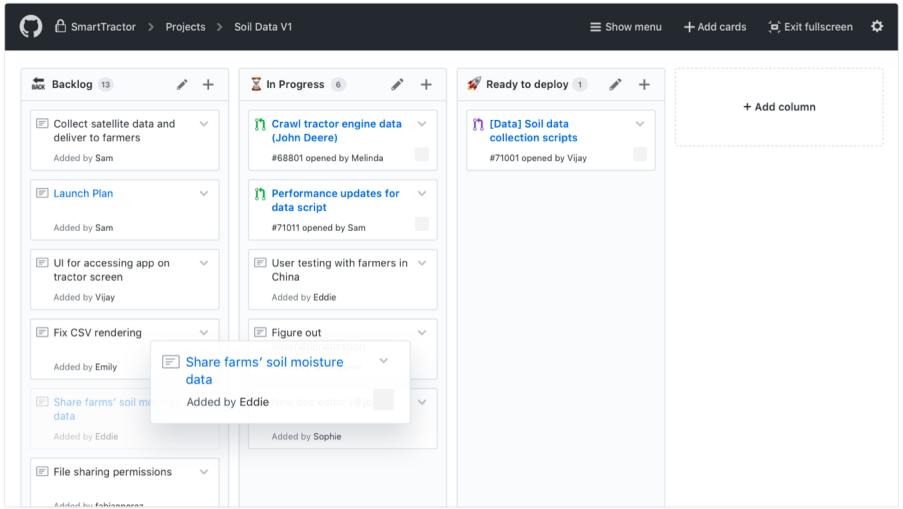
- Products:
  - Product Backlog
  - Sprint Backlog

- Process:
  - Sprint Planning Meeting
  - Daily Scrum Meeting
  - Sprint Retrospective
  - Sprint Review Meeting

Product Backlog/Sprint Backlog

- The product backlog is all the features for the product
- The sprint backlog is all the features that will be worked on for that sprint. These should be broken down into discrete tasks:
  - Fine-grained
  - Estimated
  - Assigned to individual team members
  - Acceptance criteria should be defined
- User Stories are often used

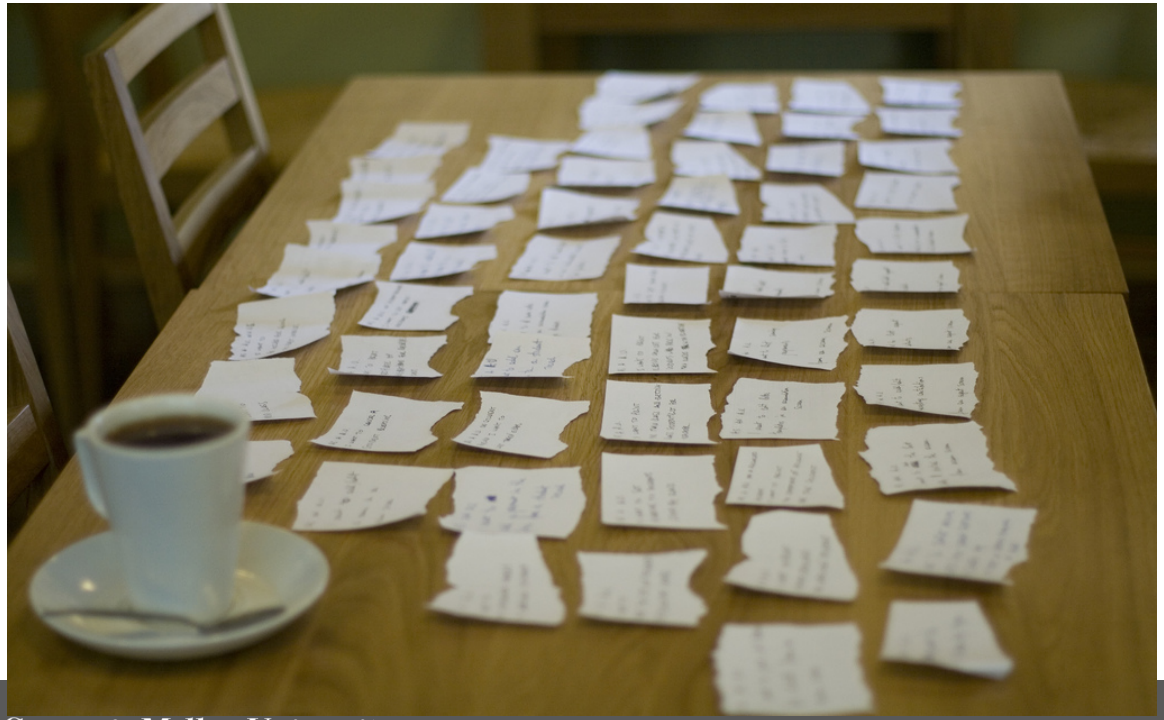
Backlog – information radiators



Scrum Meetings

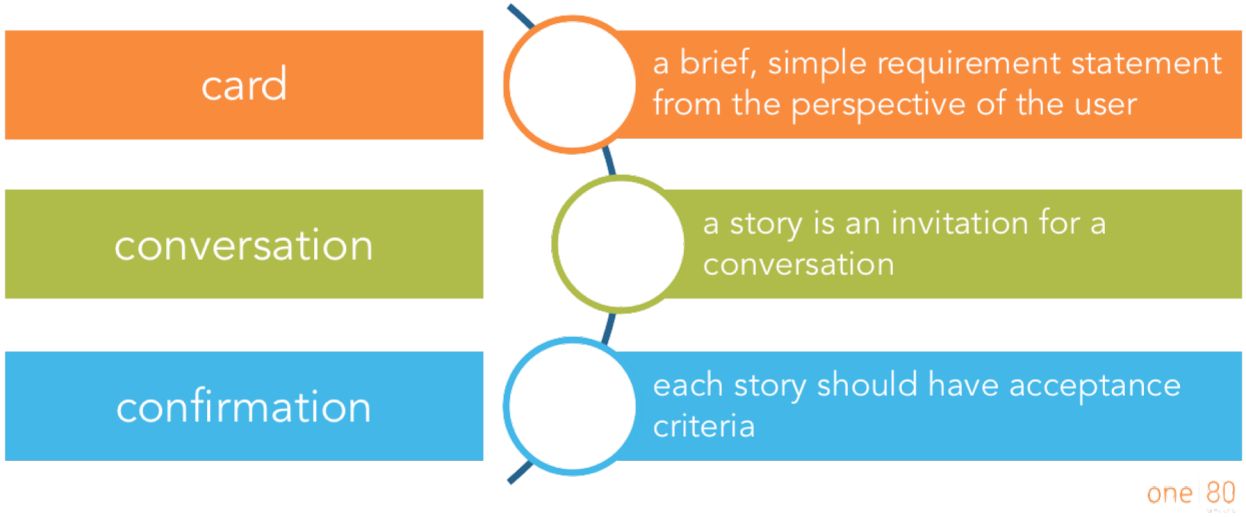
- Sprint Planning Meeting
  - Entire Team decides together what to tackle for that sprint
- Daily Scrum Meeting
  - Quick Meeting to touch base on :
    - What have I done? What am I doing next? What am I stuck on/need help?
- Sprint Retrospective
  - Review sprint process
- Sprint Review Meeting
  - Review Product

User Stories



Source: https://www.flickr.com/photos/jakuza/2728096478

User Stories



Source: http://one80services.com/user-stories/writing-good-user-stories-hint-its-not-about-writing/

The card

- “As a [role], I want [function], so that [value]”
- Should fit on a 3x5 card

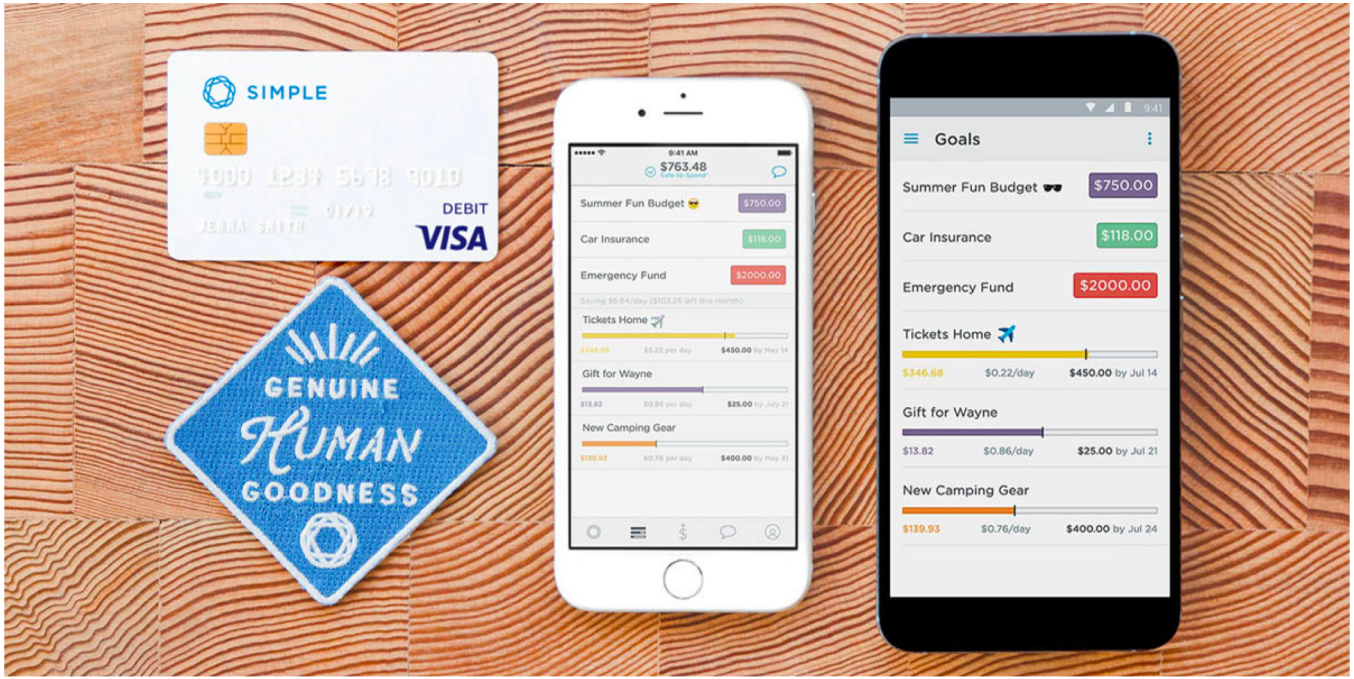
The conversation

- An open dialog between everyone working on the project and the client
- Split up Epic Stories if needed

The Confirmation

- A confirmation criteria that will show when the task is completed
- Could be automated or manual

Exercise


SIMPLE
DEBIT
VISA
GENUINE
HUMAN
GOODNESS
$763.48
Summer Fun Budget
$750.00
Car Insurance
$118.00
Emergency Fund
$2000.00
Tickets Home
$450.00 by May 14
Gift for Wayne
$25.00 by July 21
New Camping Gear
$400.00 by May 31
Goals
Summer Fun Budget
$750.00
Car Insurance
$118.00
Emergency Fund
$2000.00
Tickets Home
$346.68
$0.22/day
$450.00 by Jul 14
Gift for Wayne
$13.82
$0.86/day
$25.00 by Jul 21
New Camping Gear
$139.93
$0.76/day
$400.00 by Jul 24

How to evaluate user story?

Follow the INVEST guidelines for good user stories!

one 80

Source: http://one80services.com/user-stories/writing-good-user-stories-hint-its-not-about-writing/

Independent

- Schedule in any order.
- Not overlapping in concept
- Not always possible

Negotiable

- Details to be negotiated during development
- Good Story captures the essence, not the details

Valuable

- This story needs to have value to someone (hopefully the customer)
- Especially relevant to splitting up issues

Estimable

- Helps keep the size small
- Ensure we negotiated correctly
- “Plans are nothing, planning is everything” -Dwight D. Eisenhower

Small

- Fit on 3x5 card
- At most two person-weeks of work
- Too big == unable to estimate

Testable

- Ensures understanding of task
- We know when we can mark task “Done”
- Unable to test == do not understand

Activity

Follow the INVEST guidelines for good user stories!

one 80

TEAMWORK (STUDENT TEAMS)

(MORE ON TEAMS IN REAL PROJECTS LATER IN THE COURSE)

Expectations

- Meet initially and then regularly
- Review team policy
- Divide work and integrate
- Establish a process
- Set and document clear responsibilities and expectations
  - Possible Roles: Coordinator, Scribe, Checker, Monitor
  - Rotate roles every assignment
- Every team member should understand the entire solution

Team Policies

- see document

- Make agreements explicit and transparent
- Most teams will encounter some problem

Dealing with problems

- Openly report even minor team issues in individual part of assignments
- In-class discussions and case studies
- Additional material throughout semester
- We will attend one team meeting

Planning and In-Team Communication

- Asana, Trello, Microsoft Project, …
- Github Wiki, Google docs, …
- Email, Slack, Facebook groups, …

Homework 2 Discussion time

Further Reading

- McConnell. Software Project Survival Guide. Microsoft Press 1998, Chapter 3
- Sommerville. Software Engineering. 8th Edition. Addison-Wesley 2007. Chapters 5 "Project Planning" and 26 "Software Cost Estimation"