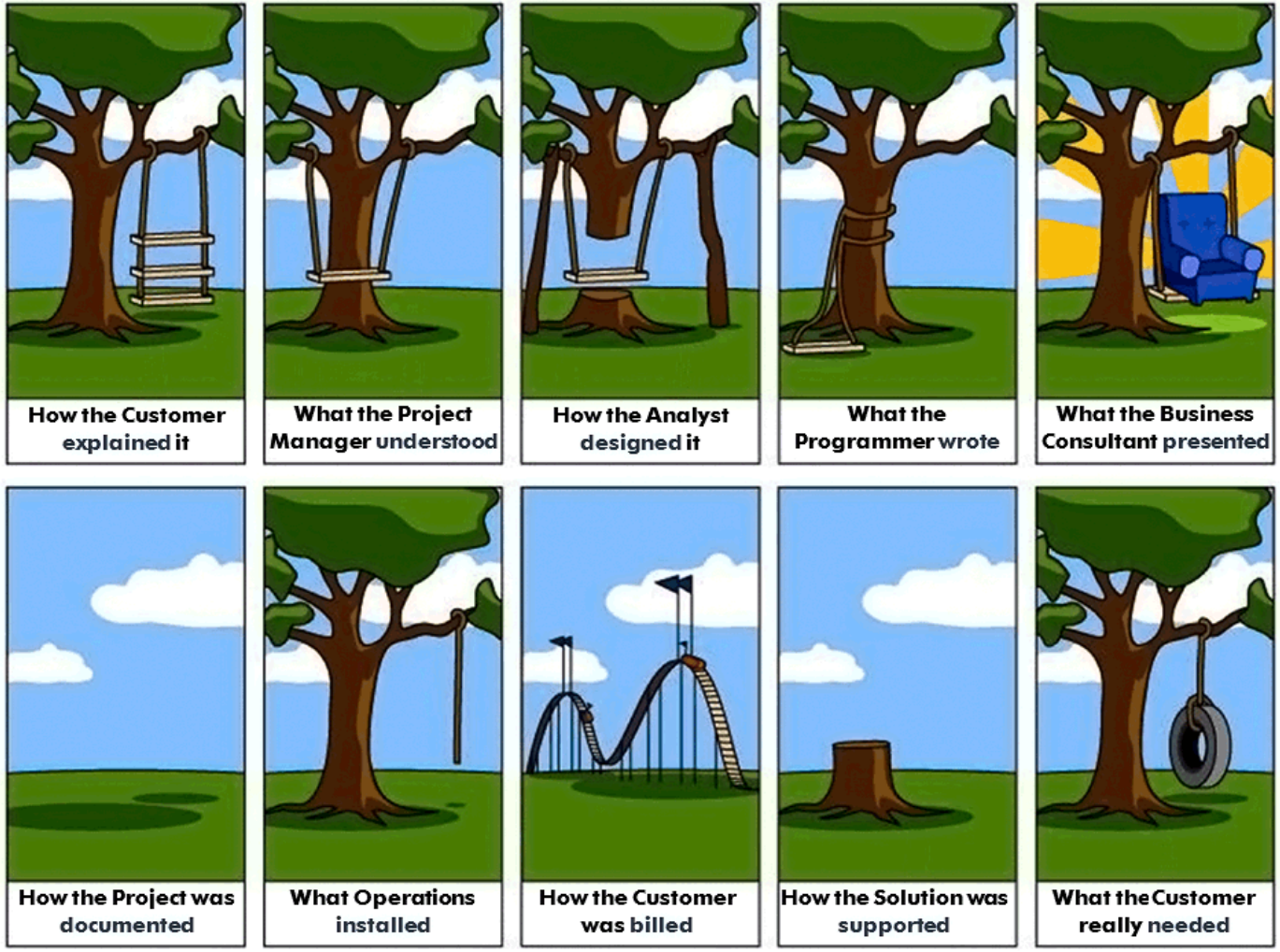

How the Customer explained it
What the Project Manager understood
How the Analyst designed it
What the Programmer wrote
What the Business Consultant presented
How the Project was documented
What Operations installed
How the Customer was billed
How the Solution was supported
What the Customer really needed

Carnegie Mellon University

Lecture 4: Intro To Process Milestones, Estimation, Planning

17-313 Fall 2023

Learning Goals

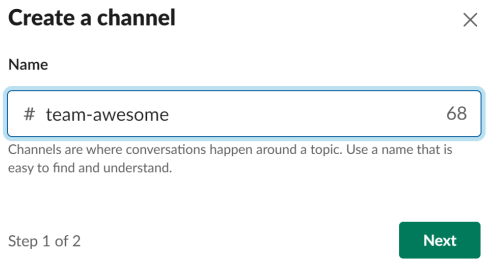
- Recognize the importance of process
- Identify why software development has project characteristics
- Understand backlogs and user stories
- Use milestones for planning and progress measurement
- Understand the difficulty of measuring progress

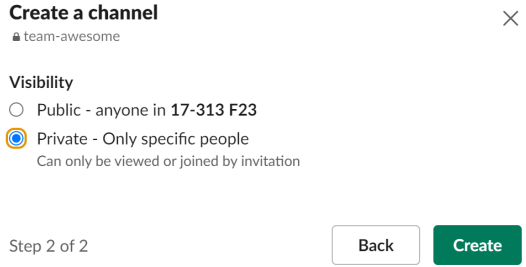
Administrivia

- Project 1 due tomorrow
- Do look closely at the guidelines (e.g., for issues, pull requests, etc) provided and follow them accordingly.
- At this point, if you cannot successfully translate the file you have chosen, it would be wise to proceed to complete the rest of the project, including making the pull request and doing the written assignment. A proper understanding of the process is also a major part of the assessment.
- For students who were successful with the implementation, it would be wise to ensure that you have followed the guidelines around the process to ensure you get full points.
- We will give partial credit for partially correct solutions. If you are turning in a PR with a partially correct conversion, please explicitly list in the text of the PR what you did successfully and what issues are outstanding that you could not fix.

Project 2

- Project 2 released on the course website tomorrow.
- Teams will be released as well.
- Extra credit: Team activity
  - Create private channel on Slack
  - Invite course staff to claim credit





Outline

- Software Process and why we need one
- Software Process Models
- Scrum
- Task and progress estimation

S3D

Outline

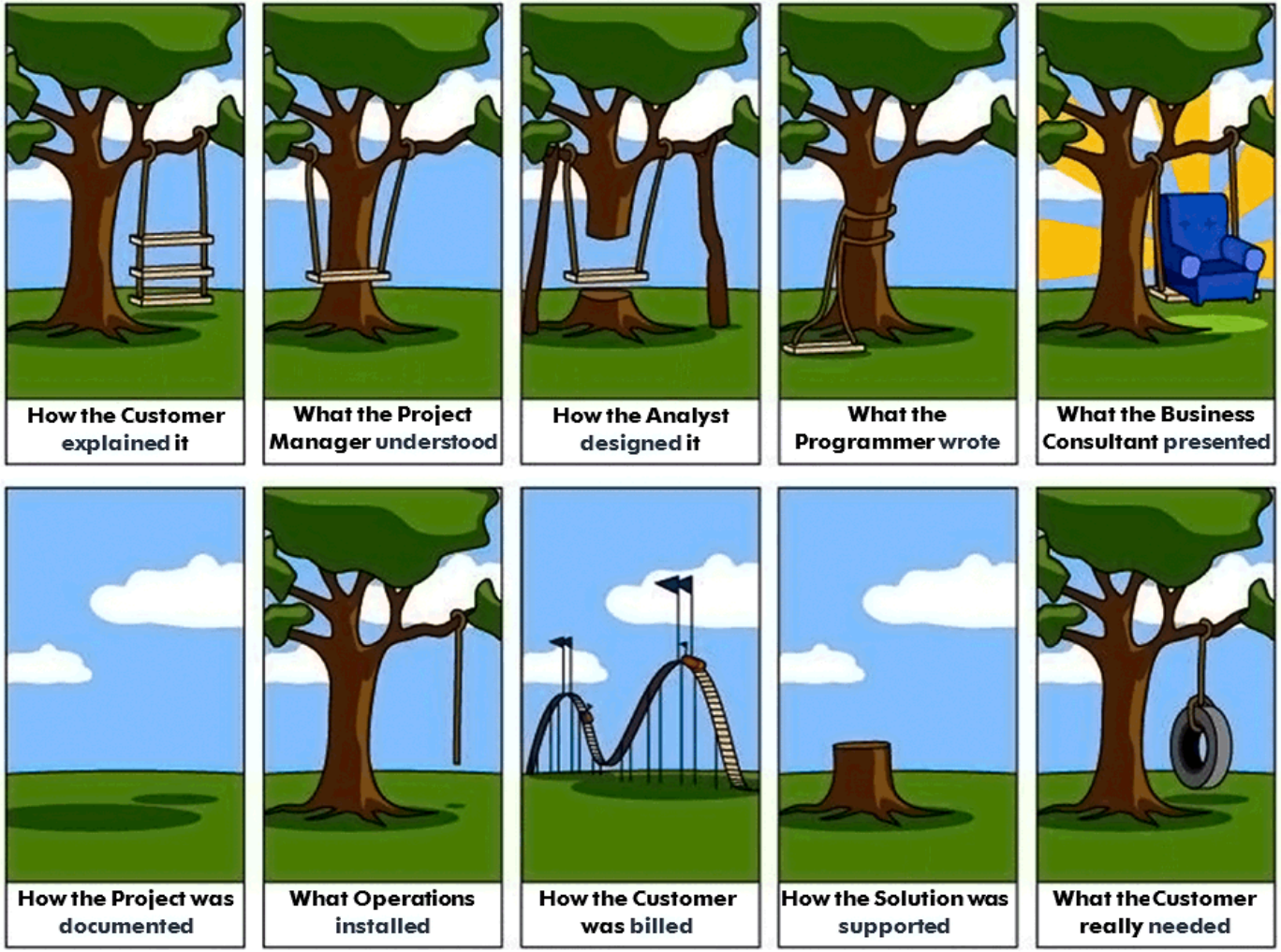
- **Software Process and why we need one**
- Software Process Models
- Scrum
- Task and progress estimation

S3D

Software Process

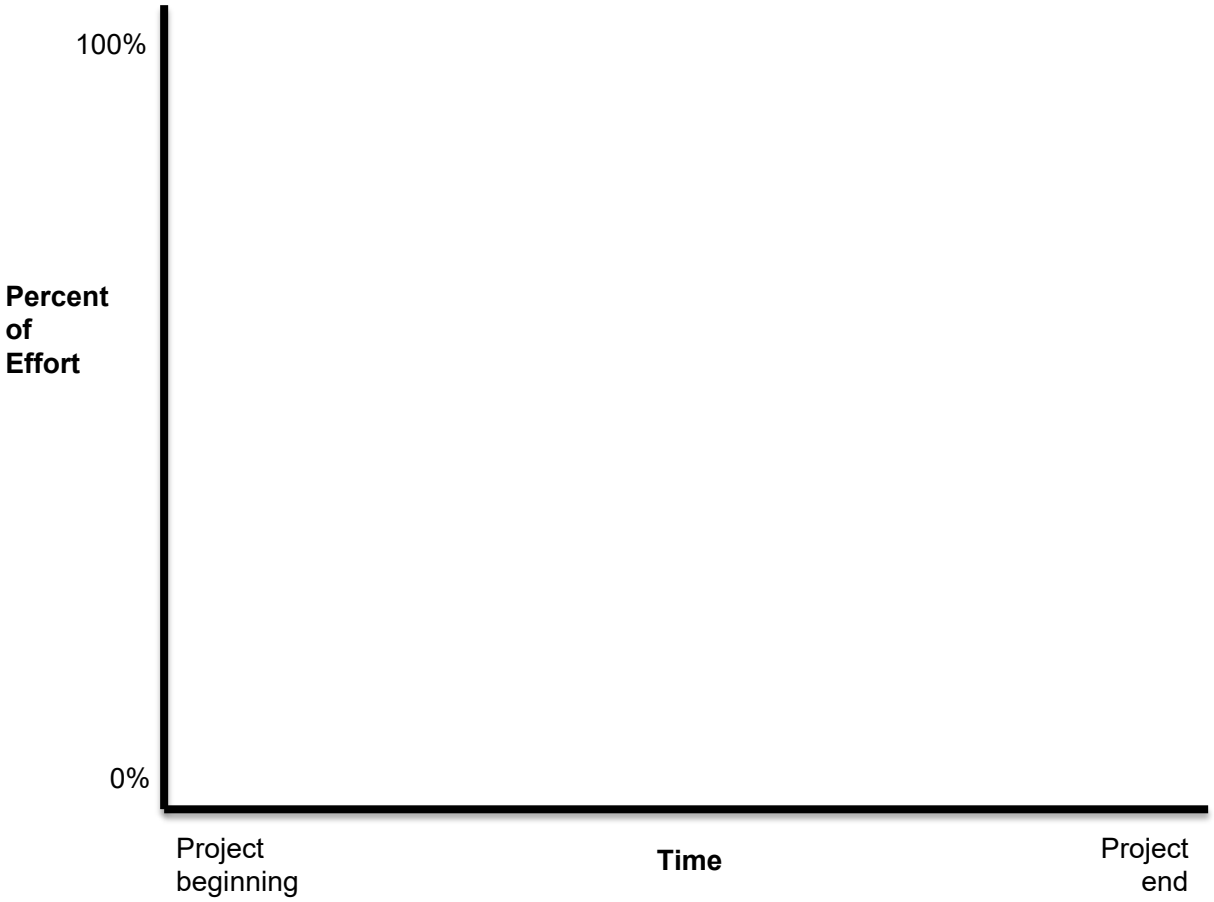
“The set of activities and associated results that produce a software product”

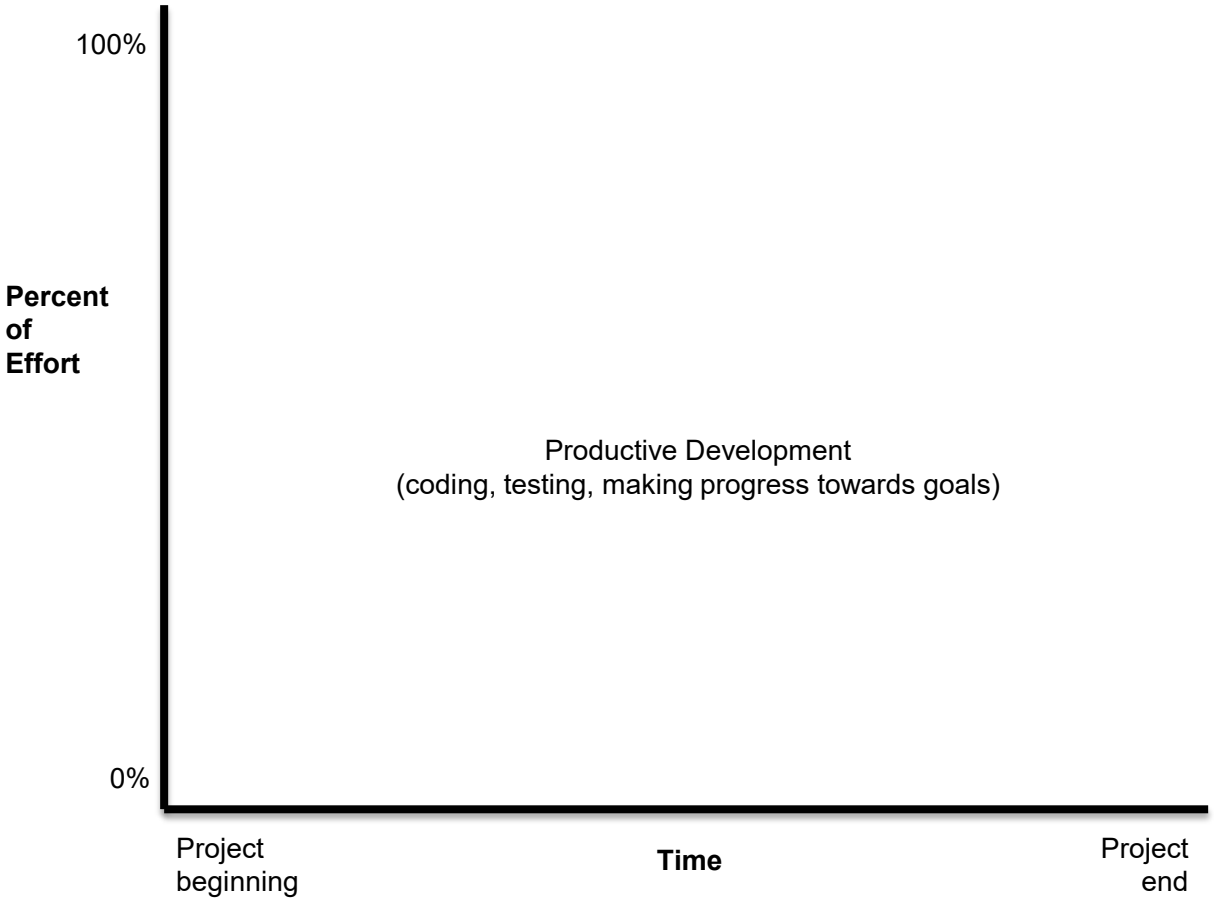
Sommerville, SE, ed. 8


How the Customer explained it
What the Project Manager understood
How the Analyst designed it
What the Programmer wrote
What the Business Consultant presented
How the Project was documented
What Operations installed
How the Customer was billed
How the Solution was supported
What the Customer really needed

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


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


100%
Percent of Effort
Productive Development
(coding, testing, making progress towards goals)
0%
Project beginning
Time
Project end

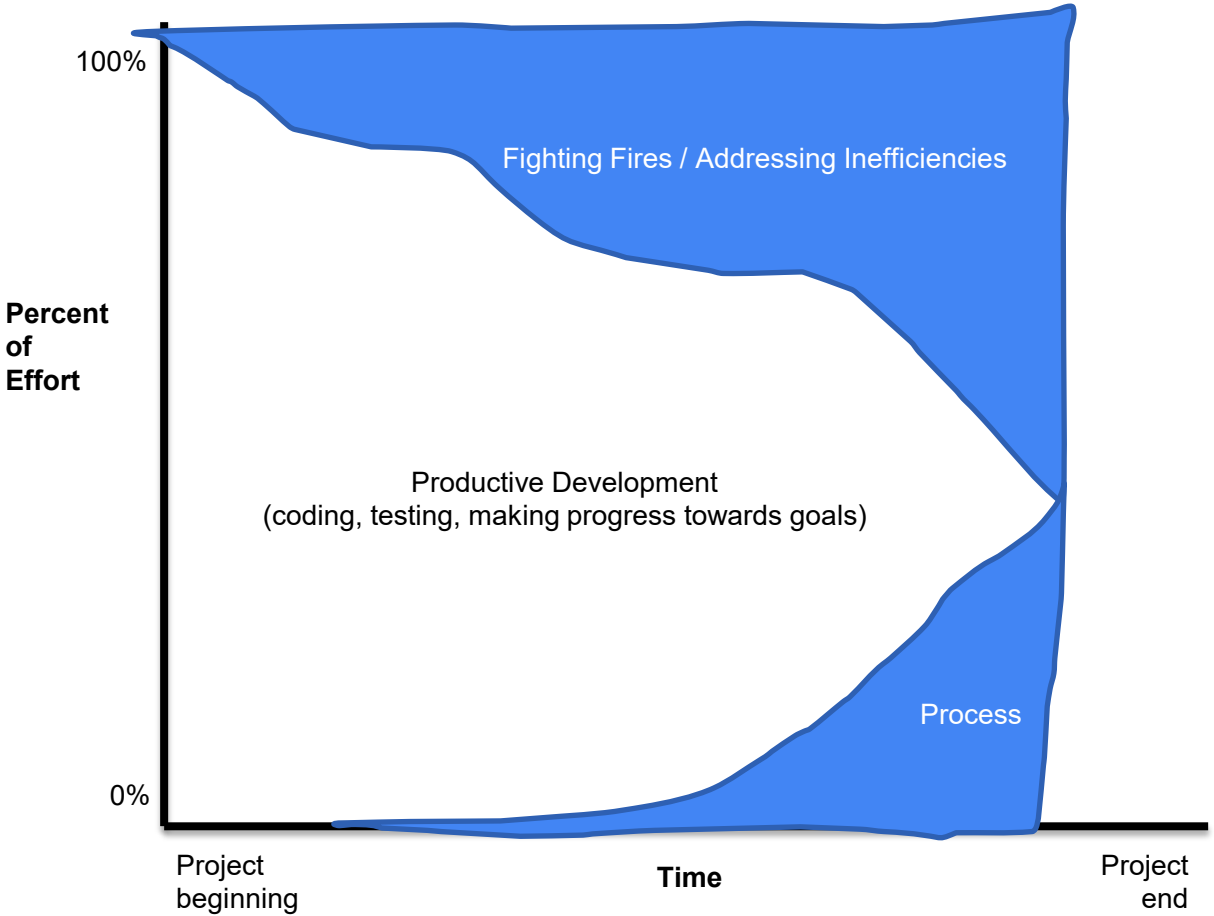

100%
Addressing Inefficiencies
Percent of Effort
Productive Development
(coding, testing, making progress towards goals)
0%
Project beginning
Time
Project end

Your manager asks you to follow a process

- Writing down all requirements
- Require approval for all changes to requirements
- Use version control for all changes
- Track all reported bugs
- Review requirements and code
- Break down development into smaller tasks and schedule and monitor them
- Planning and conducting quality assurance
- Have daily status meetings
- Use Docker containers to push code between developers and operation


100%
Addressing Inefficiencies
Percent of Effort
Productive Development
(coding, testing, making progress towards goals)
Process: Cost and Time estimates, Writing Requirements, Design,
Change Management, Quality Assurance Plan,
Development and Integration Plan
0%
Project beginning
Time
Project end

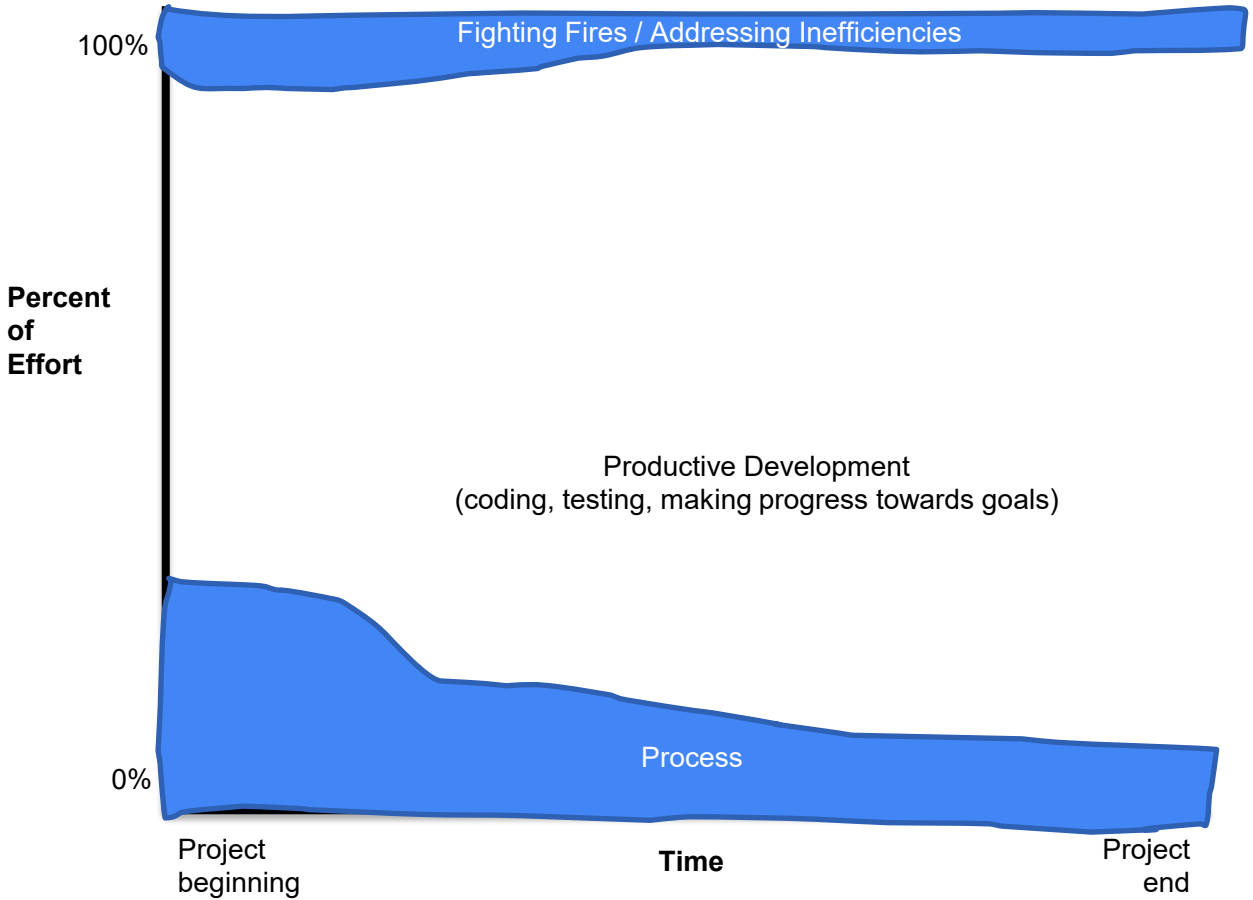

ACCEPT JOB AS SENIOR ENGINEER
8 HOURS OF MEETINGS A DAY
Posted in r/ProgrammerHumor
reddit


100%
Fighting Fires / Addressing Inefficiencies
Percent
of
Effort
Productive Development
(coding, testing, making progress towards goals)
Process
0%
Project
beginning
Time
Project
end

Example process issues

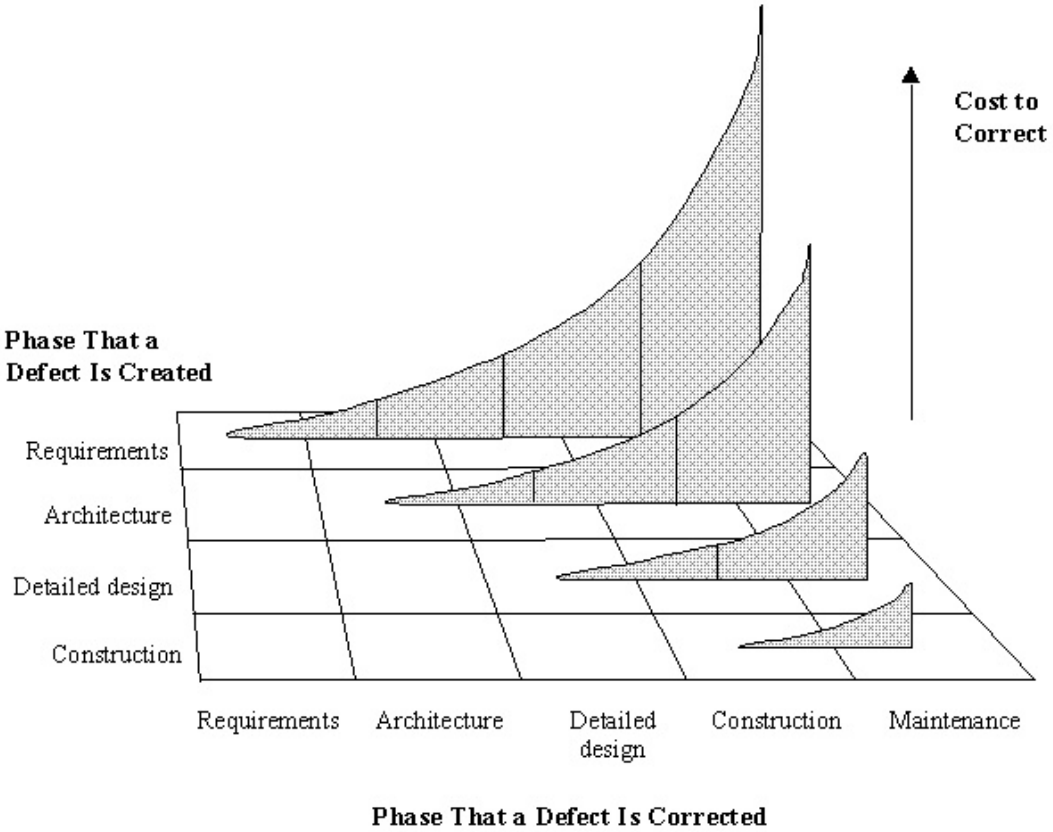
- Change Control: Mid-project informal agreement to changes suggested by customer or manager. Project scope expands 25-50%
- Quality Assurance: Late detection of requirements and design issues. Test-debug-reimplement cycle limits development of new features. Release with known defects.
- Defect Tracking: 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.

S3D



Hypothesis: Process increases flexibility and efficiency

Ideal Curve: Upfront investment for later greater returns



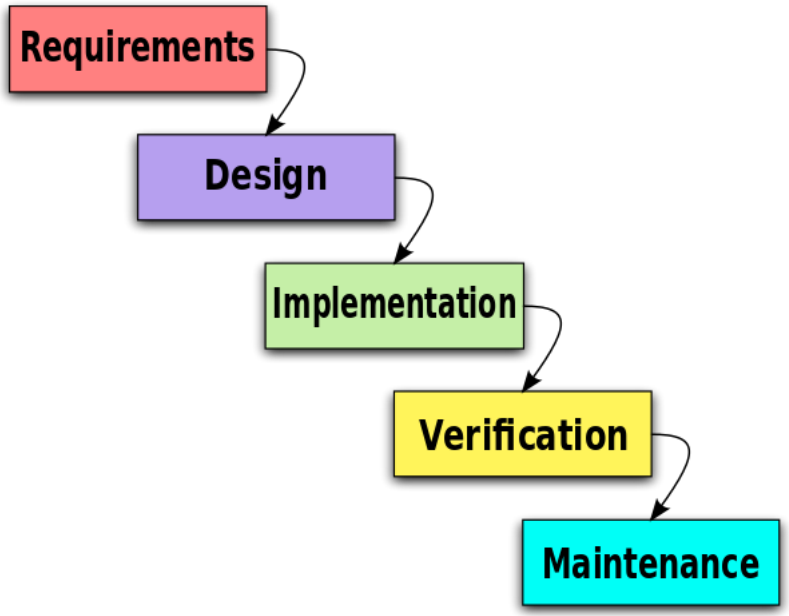
Outline

- Software Process and why we need one
- ➢ **Software Process Models**
- Scrum
- Task and progress estimation

S3D

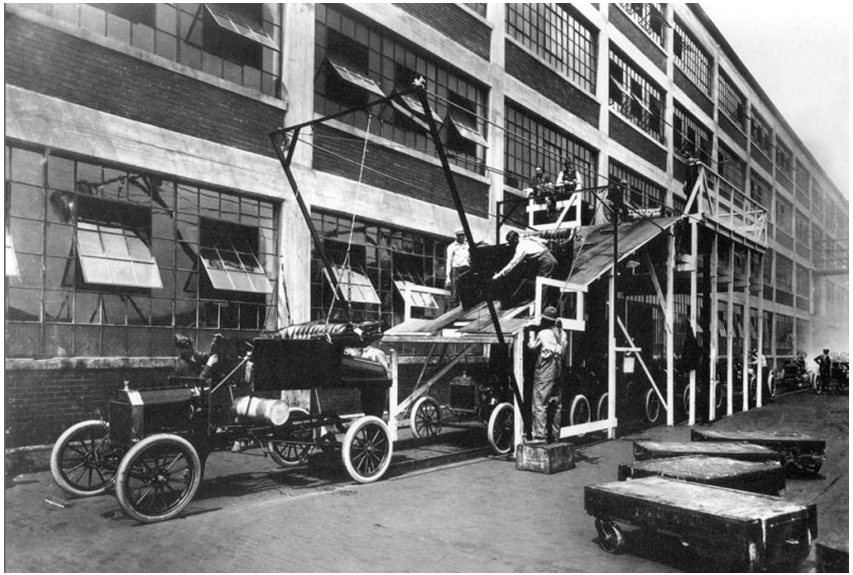
Software Process Models

Waterfall model was the original software process

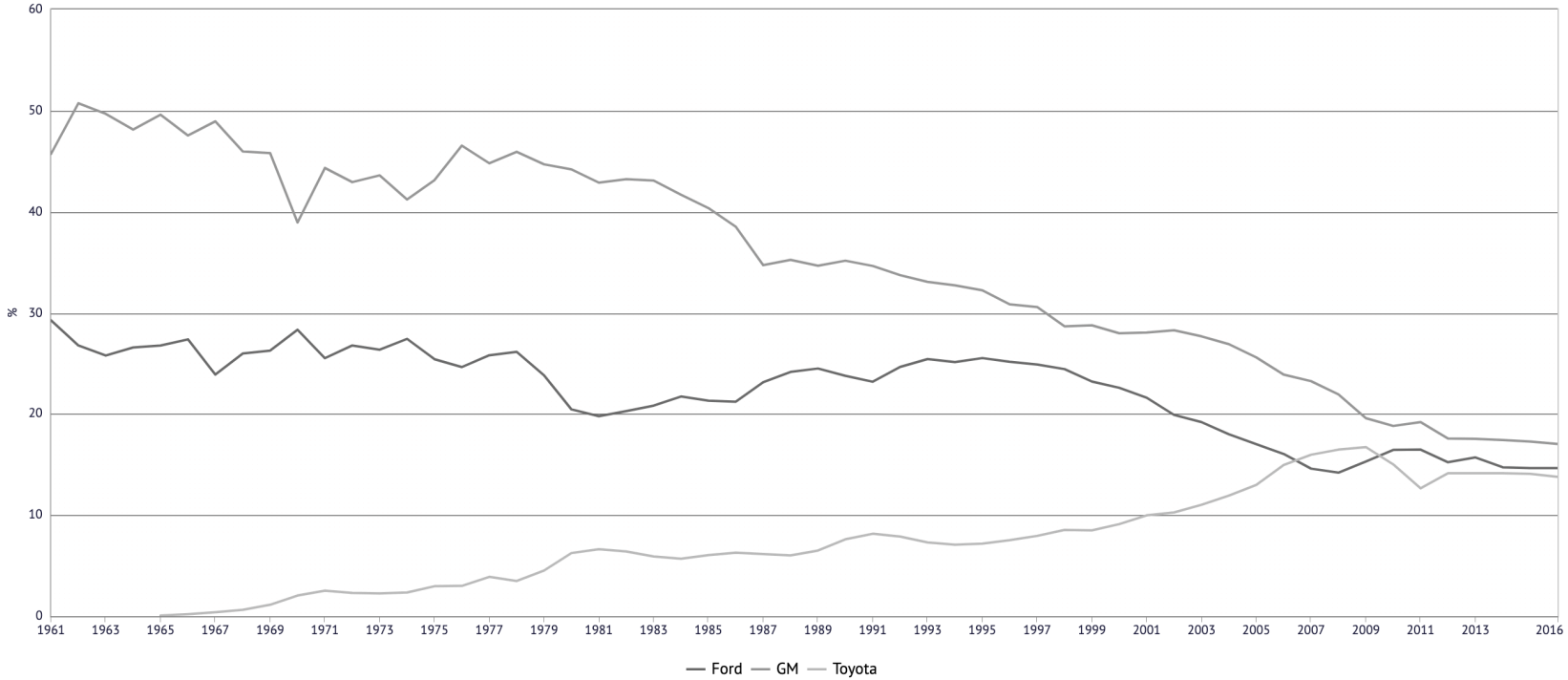


Waterfall diagram CC-BY 3.0 **Paulsmith99** at **en.wikipedia**

… akin to processes pioneered in mass manufacturing (e.g., by Ford)



US vehicle sales market share; 1961—2016 (source: knoema.com)



Lean production adapts to variable demand

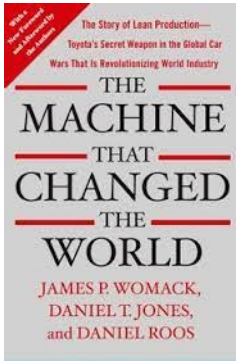
Taiichi Ohno

Toyota Production System (TPS)

- Build only what is needed, only when it is needed.
- Use the “pull” system to avoid overproduction. (Kanban)
- Stop to fix problems, to get quality right from the start (Jidoka)
- Workers are multi-skilled and understand the whole process; take ownership
- Enabling teams to have autonomy and control to change/improve quickly

Lots of software buzzwords invented recently build on these ideas

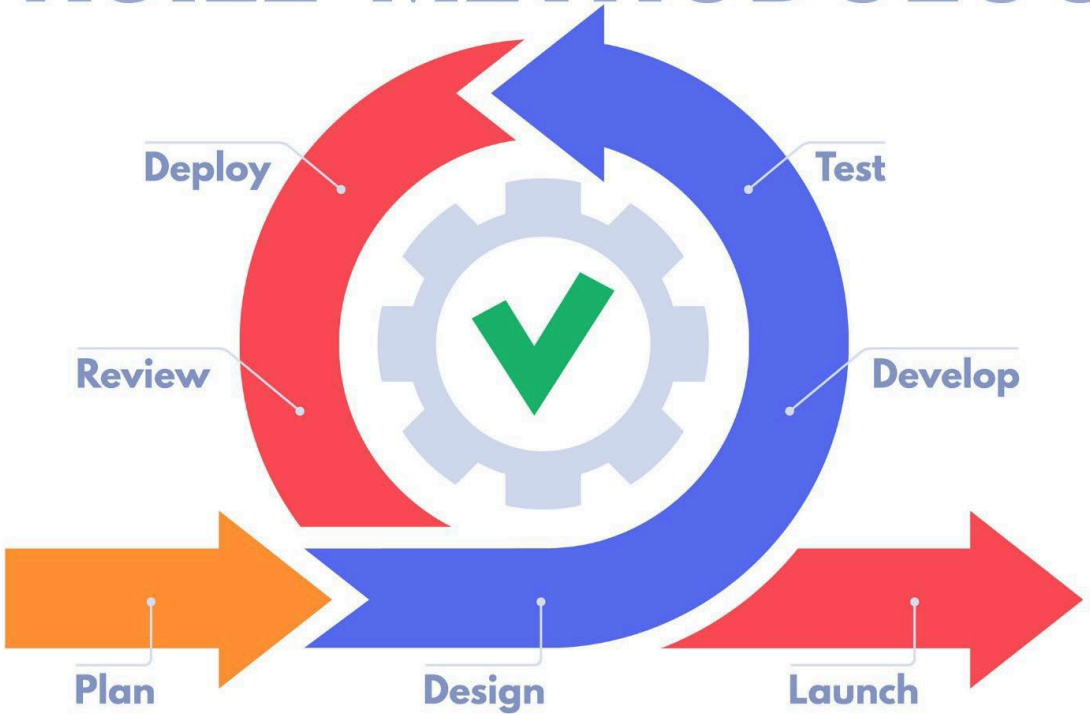
- DevOps, Shift-Left



See also: “The machine that changed the world” by James P Womack et al. The Free Press, 2007.


Kaizen cycle for continuous improvement
Kaizen requires identifying areas for improvement, creating solutions and plans for a rollout—and then cycling through the process again for other issues or issues that were inadequately addressed.
1 Get employees involved.
2 Gather a list of problems.
3 Encourage solutions, then choose an idea.
4 Test the solution.
5 Regularly measure and analyze the results.
6 If successful, adopt the solution.
7 Repeat on an ongoing basis.

AGILE METHODOLOGY



Outline

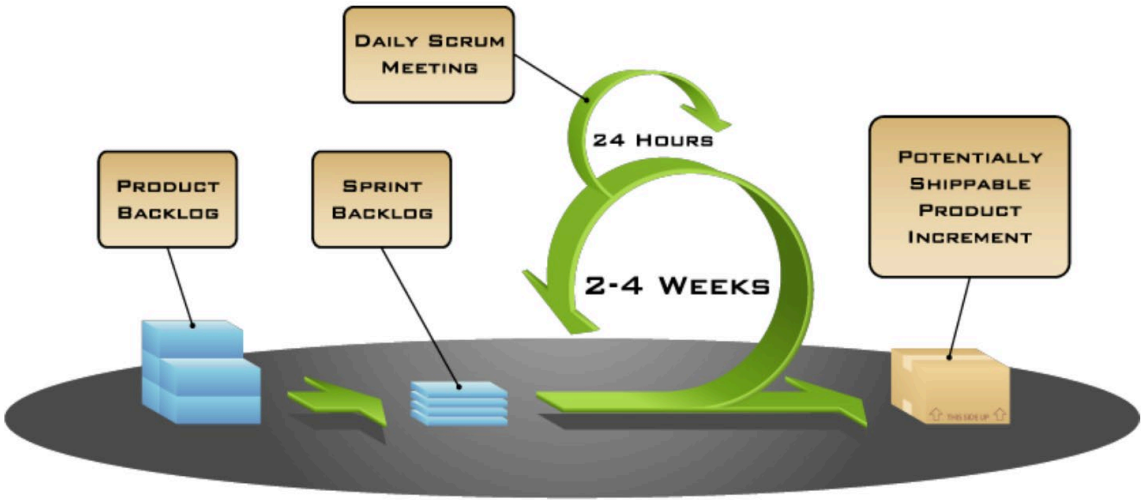
- Software Process and why we need one
- Software Process Models
- **Scrum**
- Task and progress estimation

S3D

Scrum

(Only a brief intro)

Elements of Scrum



Products:
Product Backlog
Sprint Backlog

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

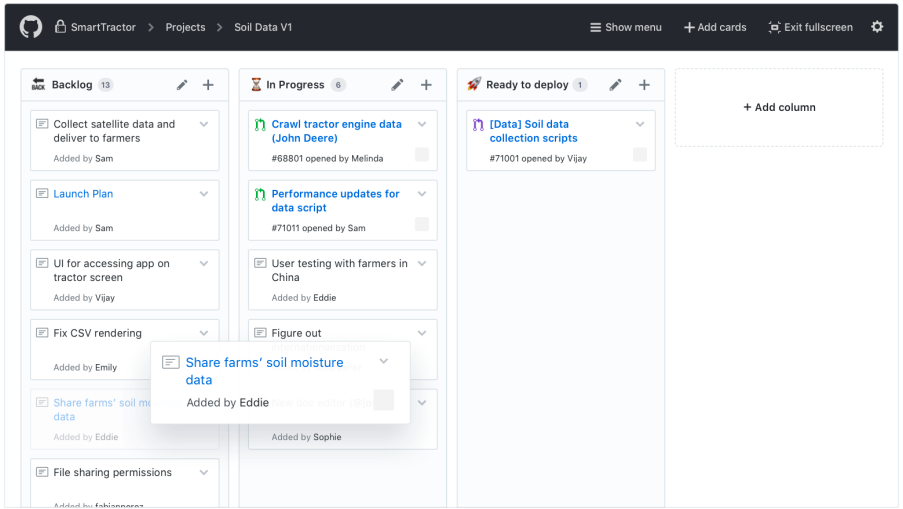
Backlogs

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

Kanban boards



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?

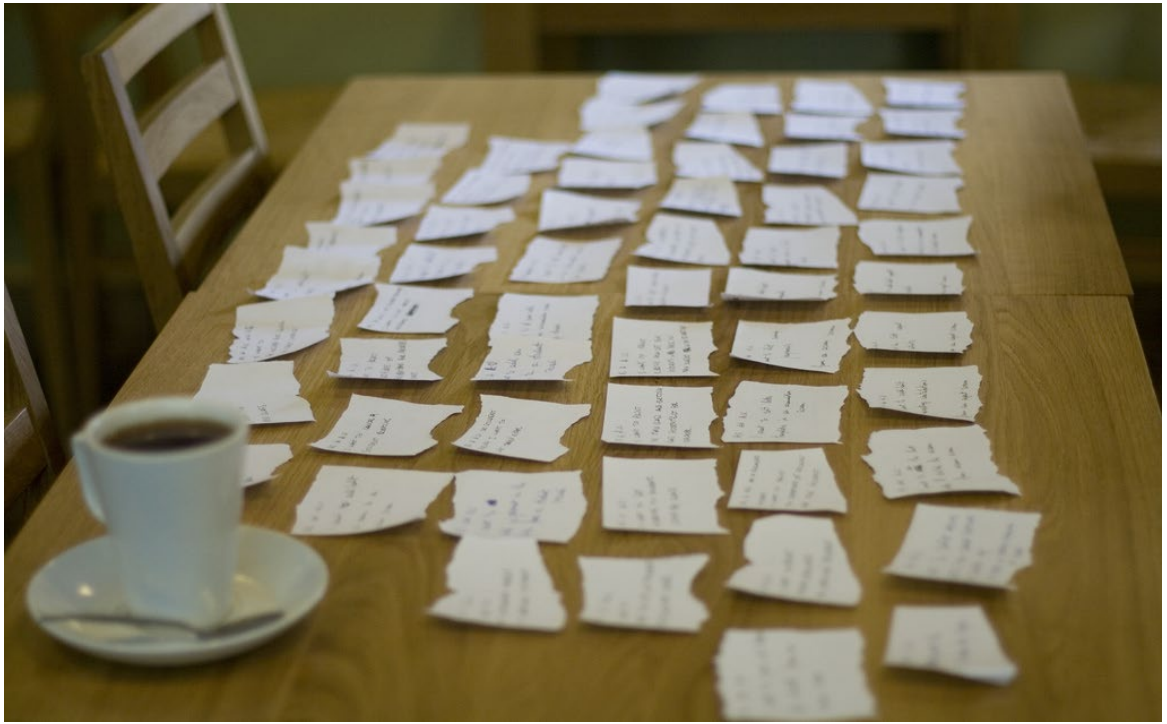
Sprint Retrospective

Review sprint process

Sprint Review Meeting

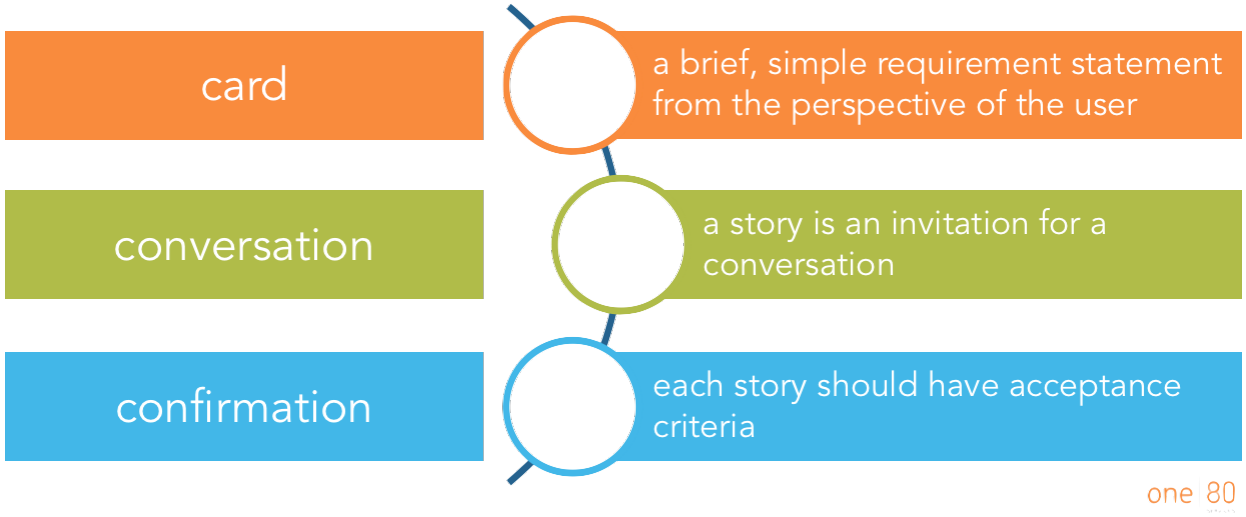
Review Product

User Stories



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

User Stories

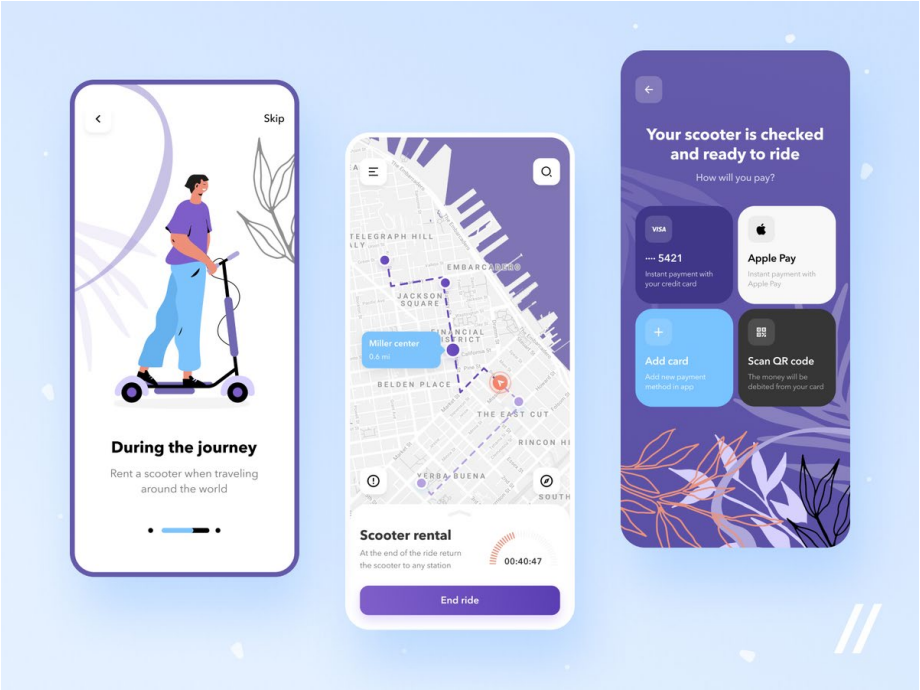


Source: http://one80services.com/user-stories/writing-good-user-stories-hint-its-not-about-writing/

User story cards (3”x5”)

“As a [role], I want [function], so that [value]”

Exercise



https://dribbble.com/shots/12512417-Scooter-Rental-App-Design

How to evaluate user story?

Follow the INVEST guidelines for good user stories!

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

one 80

Independent

- Schedule in any order.
- 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)

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 (one sprint)
- Too big == unable to estimate

Testable

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

Outline

- Software Process and why we need one
- Software Process Models
- Scrum
- ➢ **Task and progress estimation**

S3D

Time estimation

https://xkcd.com/612/

Activity: Estimate Time

Task A: Simple web version of the Monopoly board game with Doha street names

Developer Team: just you

Task B: Bank smartphone app

Developer Team: you with a 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)

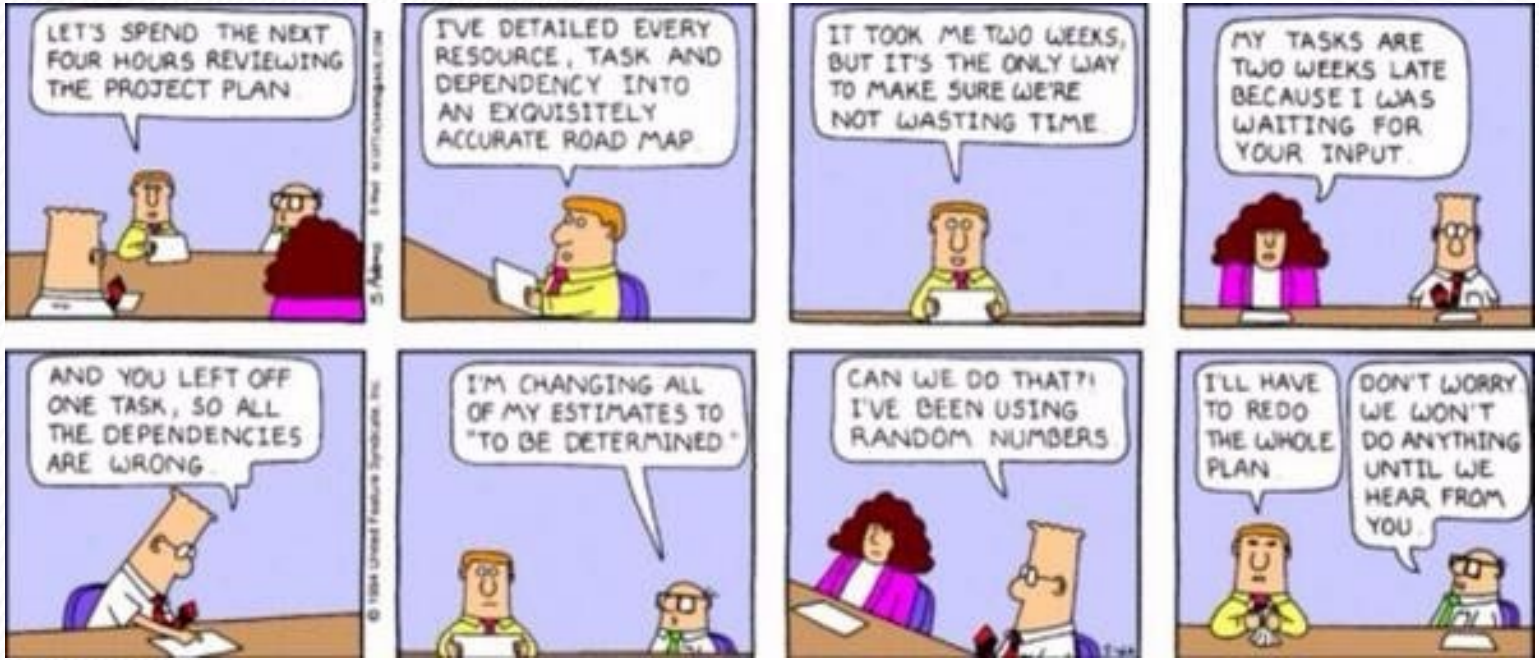
Enter your answer here:

bit.ly/313-activity-estimation

Only the number of days

Revise Time Estimate

- Do you have a comparable experience to base an estimate on?
- How much design do you need for each task?
- Break down the task into ~5 smaller tasks and estimate them.
- Revise your overall estimate if necessary


LET'S SPEND THE NEXT FOUR HOURS REVIEWING THE PROJECT PLAN
I'VE DETAILED EVERY RESOURCE, TASK AND DEPENDENCY INTO AN EXQUISITELY ACCURATE ROAD MAP
IT TOOK ME TWO WEEKS, BUT IT'S THE ONLY WAY TO MAKE SURE WE'RE NOT WASTING TIME
MY TASKS ARE TWO WEEKS LATE BECAUSE I WAS WAITING FOR YOUR INPUT
AND YOU LEFT OFF ONE TASK, SO ALL THE DEPENDENCIES ARE WRONG
I'M CHANGING ALL OF MY ESTIMATES TO "TO BE DETERMINED"
CAN WE DO THAT?! I'VE BEEN USING RANDOM NUMBERS
I'LL HAVE TO REDO THE WHOLE PLAN
DON'T WORRY WE WON'T DO ANYTHING UNTIL WE HEAR FROM YOU

Journal of Systems and Software
Volume 5, Issue 1, February 1985, Pages 49-58

Managing programming productivity

D.R. Jeffery, M.J. Lawrence

Table 5.3
Productivity by Estimation Approach (Full Result)

EFFORT ESTIMATE PREPARED BY	AVERAGE PRODUCTIVITY	NUMBER OF PROJECTS
Programmer alone	8.0	19
Supervisor alone	6.6	23
Programmer & supervisor	7.8	16
Systems analyst	9.5	21
(No estimate)	12.0	24

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 make progress *observable*

Milestone: clear end point of a (sub)tasks

- For project manager
- Reports, prototypes, completed subprojects
- "80% done" is not a suitable milestone

Deliverable: Result for customer

- Similar to milestones, but for customers
- Reports, prototypes, completed subsystems

What you need to know

- Recognize the importance of process
- Main ideas of Agile/Scrum
- Understand backlogs and user stories
- Understand the difficulty of estimating tasks and progress
- We use milestones for planning and progress measurement