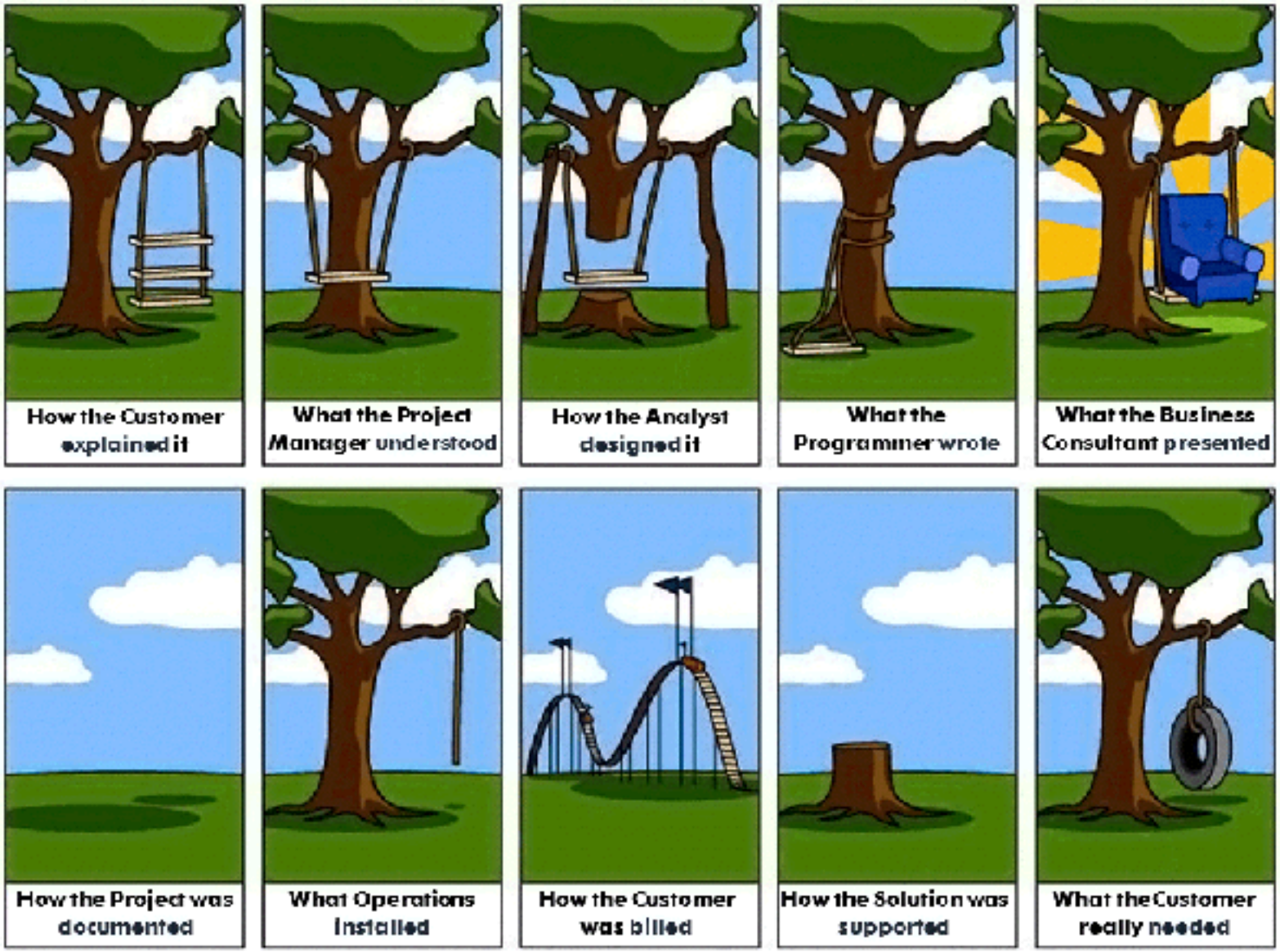

How the Customer explained it
What the Project Manager understood
How the Analyst designed it
What the Programmer wrote
What the Business Consultant presented
How the Project was documented
What Operations Installed
How the Customer was billed
How the Solution was supported
What the Customer really needed

Carnegie Mellon University

Lecture 5: Intro To Process

Milestones, Estimation, Requirements, Planning

Spring 2023

S3D
Software and Societal Systems Department

Smoking Section

- Last full row

Administrivia

- Project 1 due tonight.
- Do look closely at the guidelines (e.g. for issues, pull requests, etc) provided and follow them accordingly.
    - At this point, if you are not able to successfully translate the file you have chosen, it would be wise to proceed on to complete the rest of the project, including making the pull request and doing the written assignment. A proper understanding of the process is a major part of the assessment as well.
    - 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 were unable to fix.

S3D

Administrivia

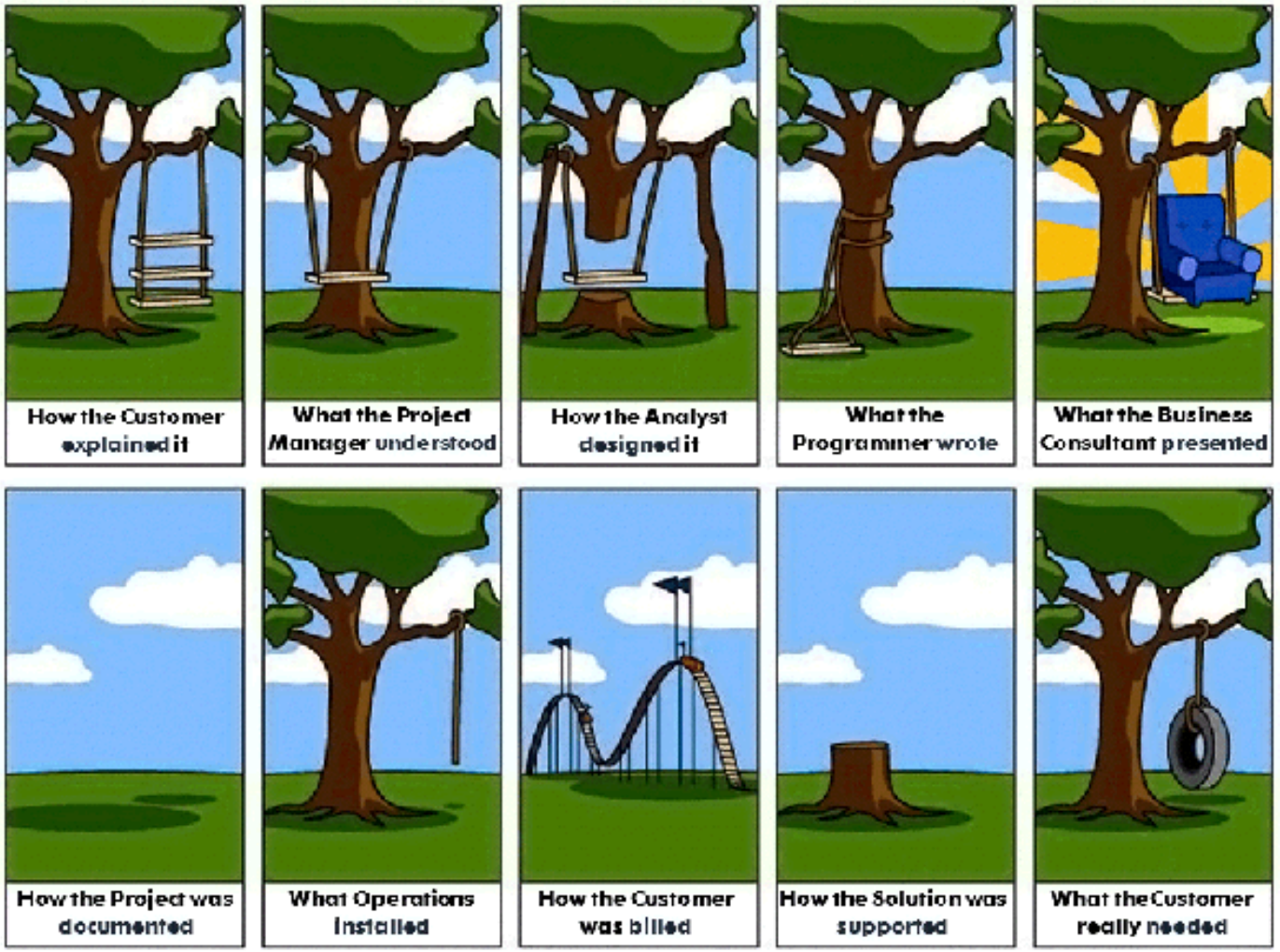
- Project 2 released on the course website tomorrow.
- Teams will be released as well.
- Extra credit: Team activity
  - Create **private** channel on Slack
  - Invite your TA mentors to claim credit

S3D

Software Process

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

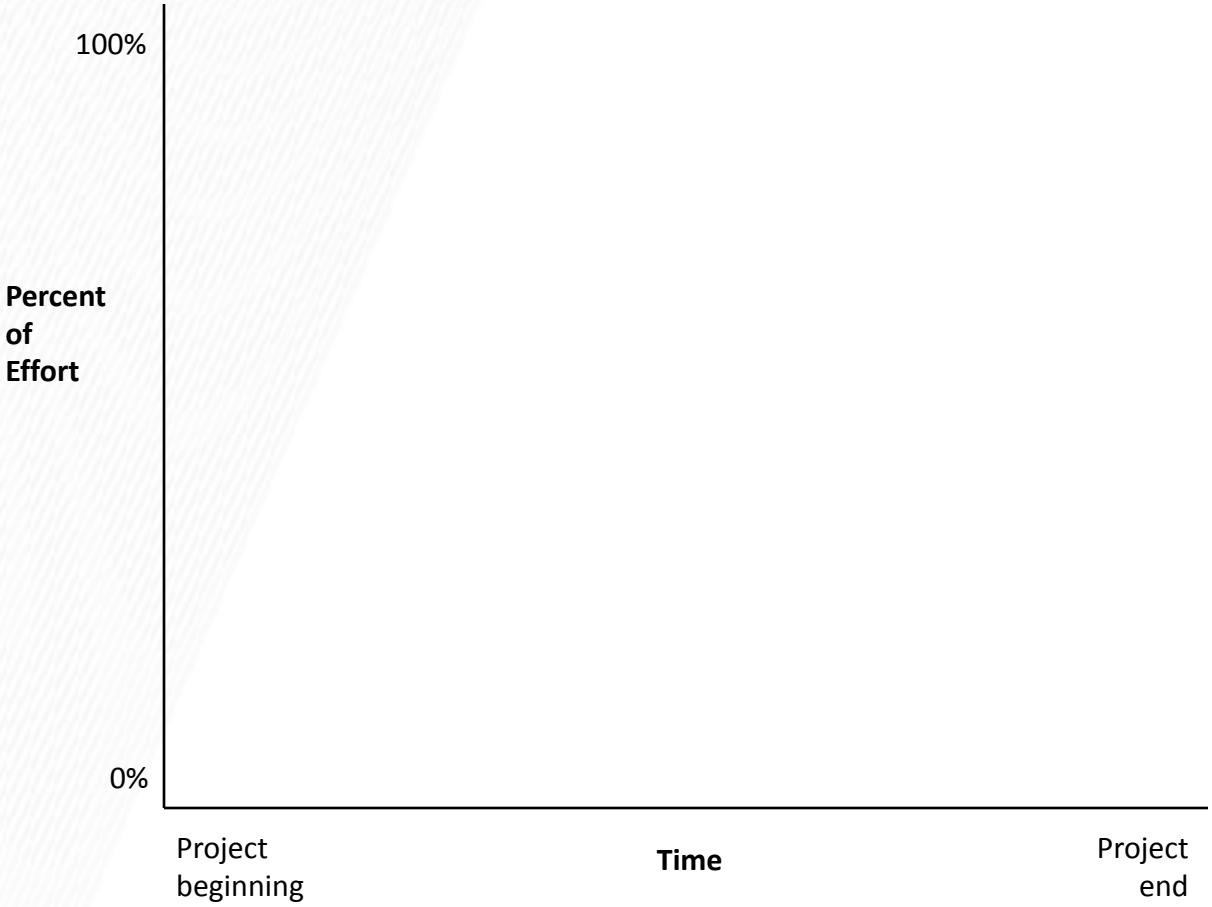
Sommerville, SE, ed. 8

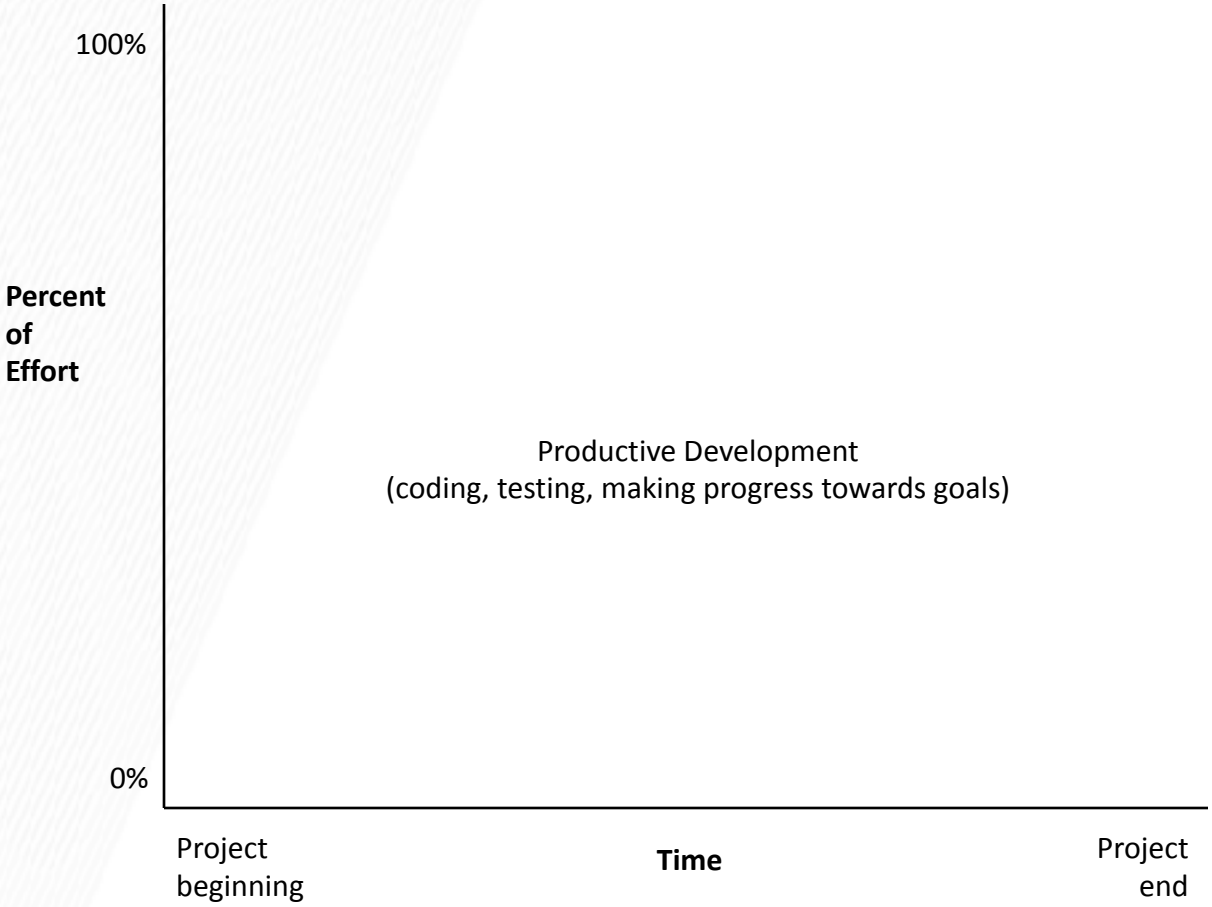

How the Customer explained it
What the Project Manager understood
How the Analyst designed it
What the Programmer wrote
What the Business Consultant presented
How the Project was documented
What Operations Installed
How the Customer was billed
How the Solution was supported
What the Customer really needed

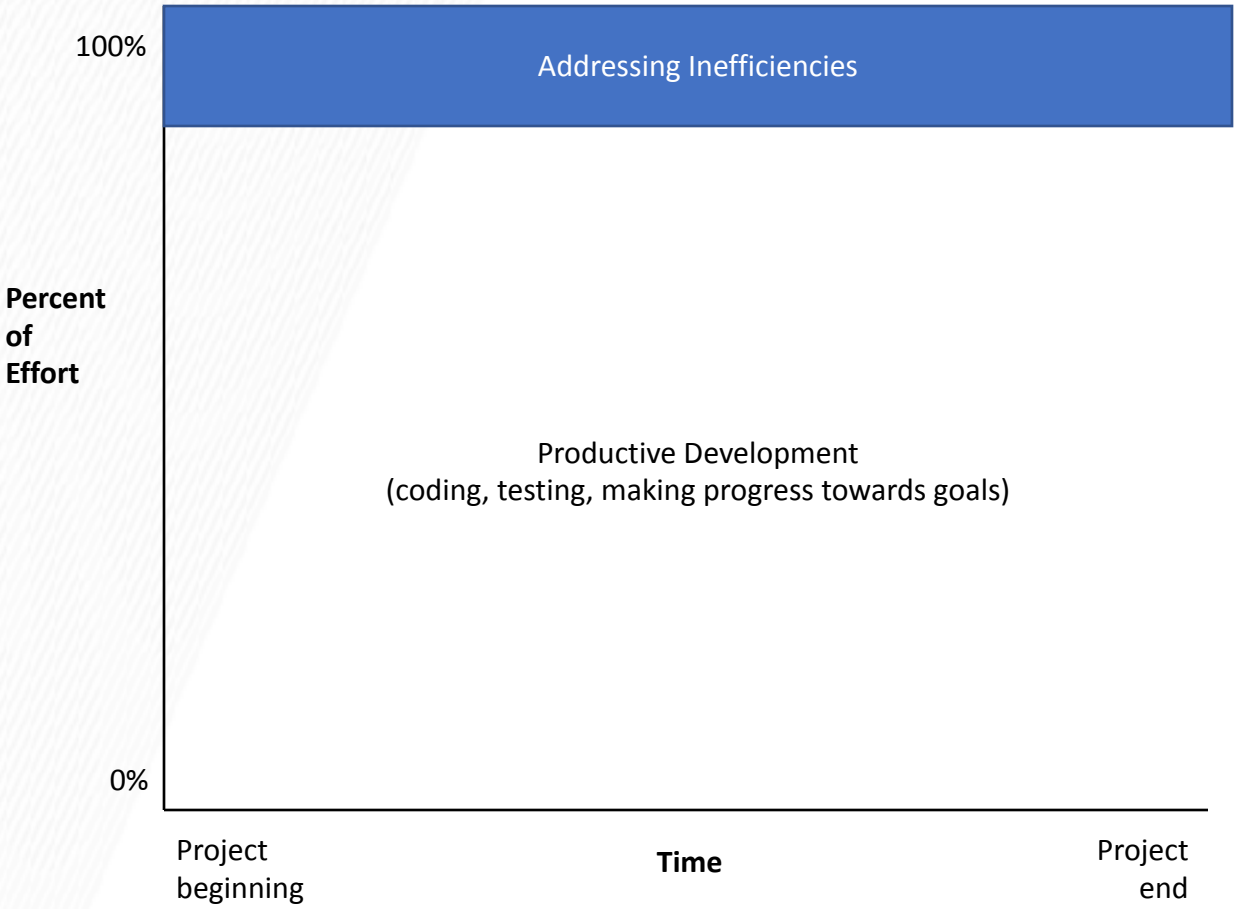
How to develop software?

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

S3D


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

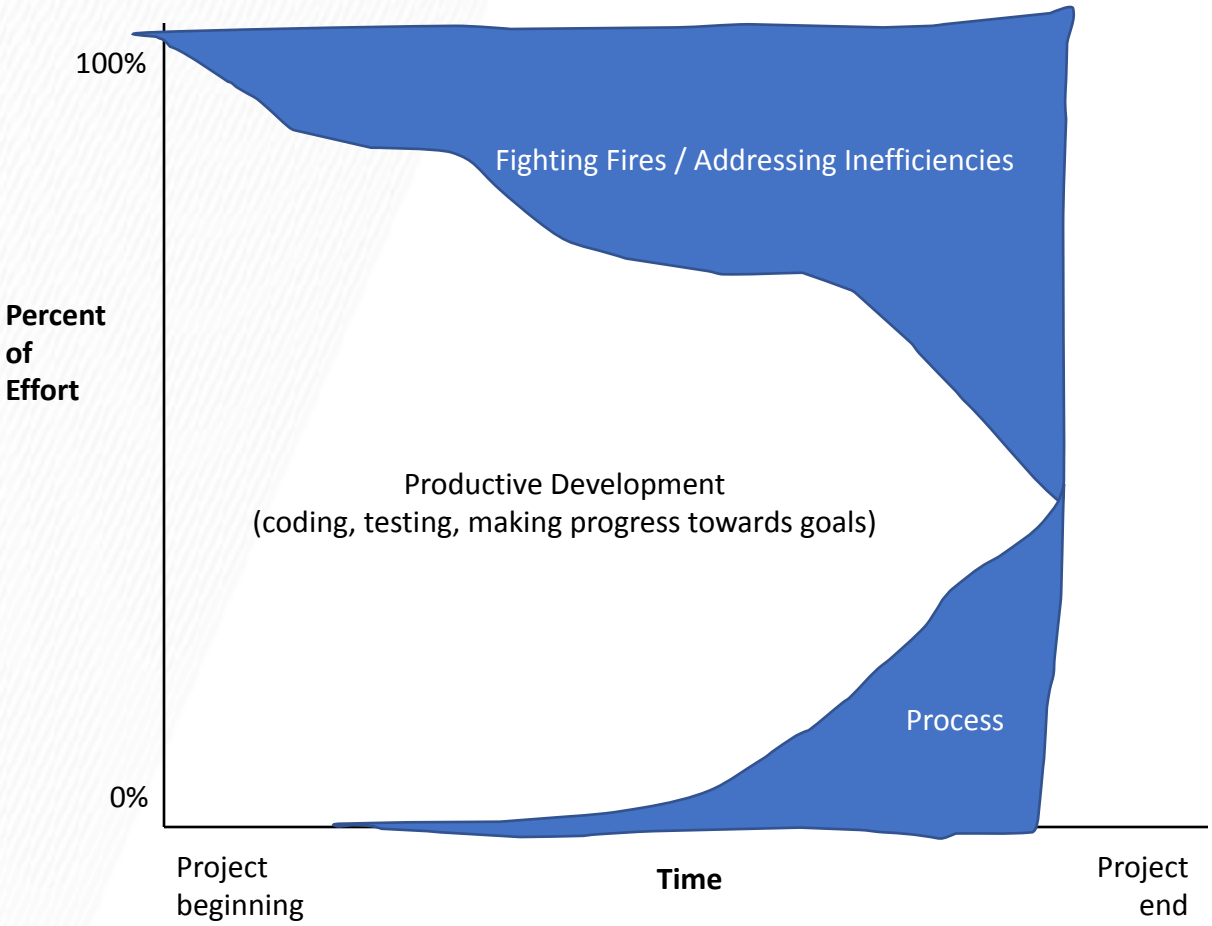

100%
Addressing Inefficiencies
Percent of Effort
Productive Development
(coding, testing, making progress towards goals)
0%
Project beginning
Time
Project end

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

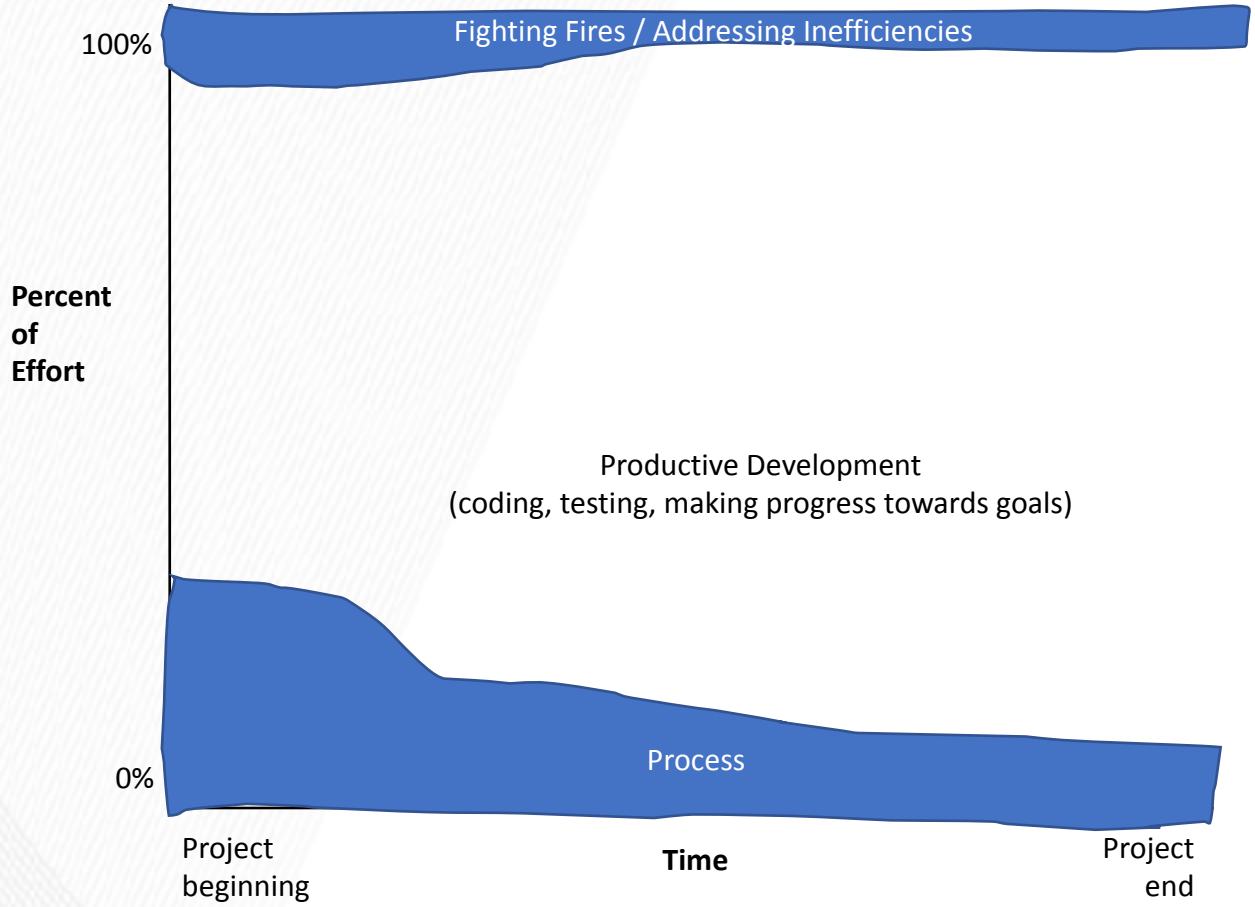
S3D


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


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

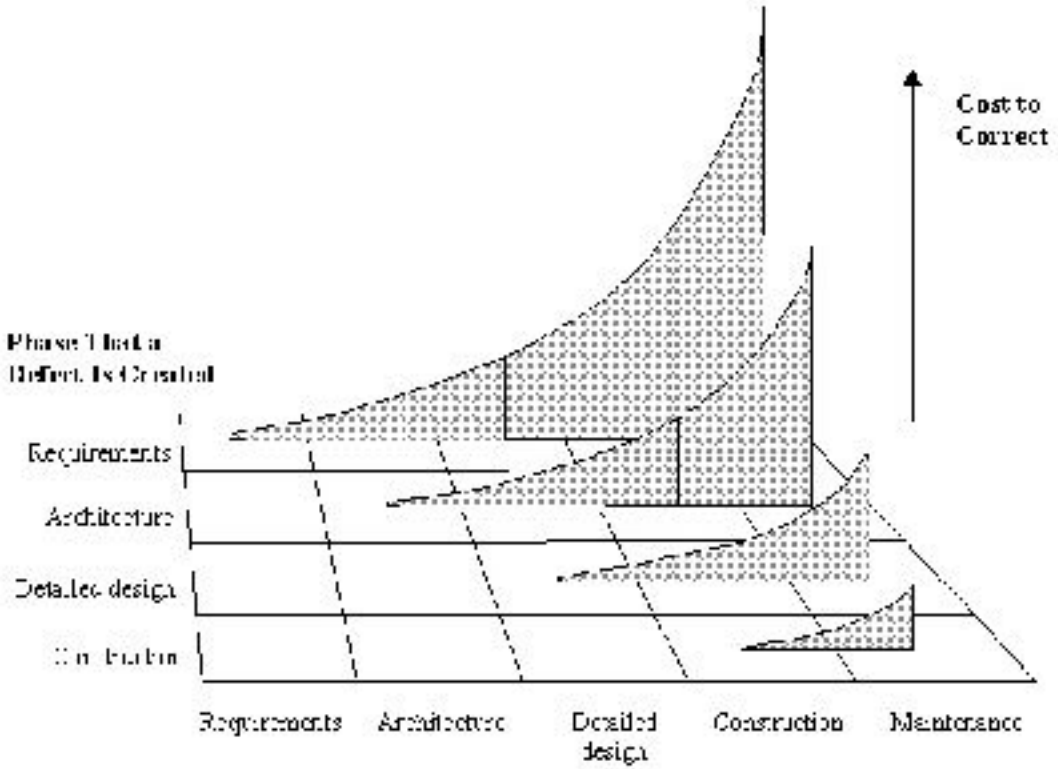
- Change Control: Mid-project informal agreement to changes suggested by customer or manager. Project scope expands 25-50%
- Quality Assurance: Late detection of requirements and design issues. Test-debug-reimplement cycle limits development of new features. Release with known defects.
- Defect Tracking: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



Hypothesis: Process increases flexibility and efficiency

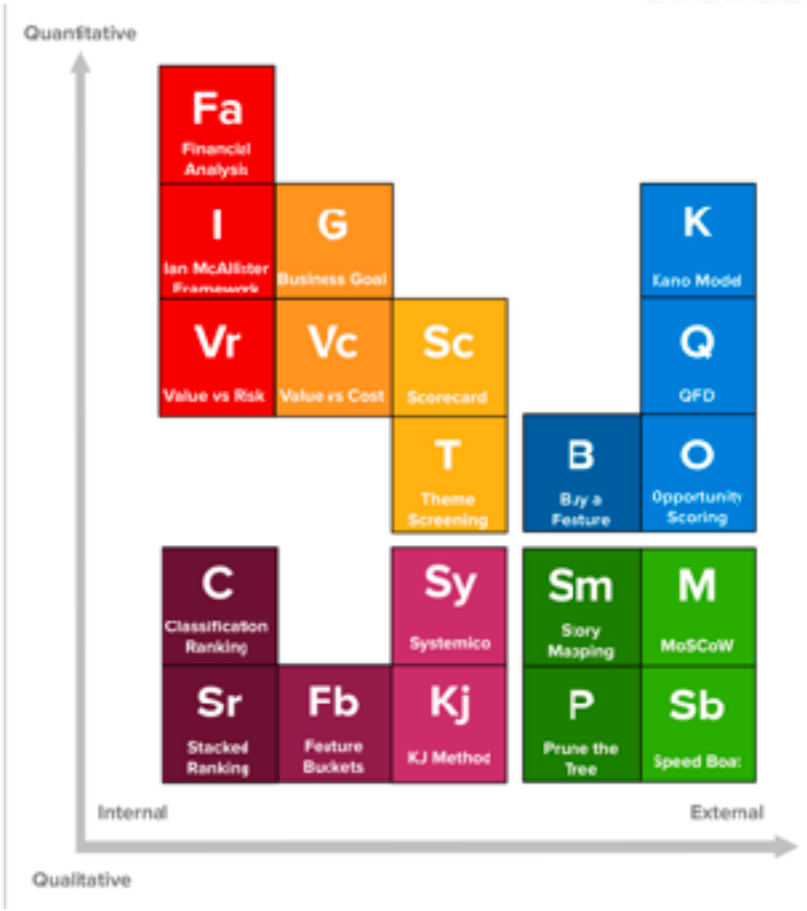
Ideal Curve: Upfront investment for later greater returns

S3D



Planning

This is a hard problem



“Periodic Table Format for Prioritization Technique”

S3D

Time estimation

https://xkcd.com/612/

S3D

Activity: Estimate Time

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

Team: just you

Task 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)

```
My Task A estimate: ___
My Task B estimate: ___


Other Task A estimate: __
Other Task B estimate: __


Other Task A estimate: __
Other Task B estimate: __
```

S3D

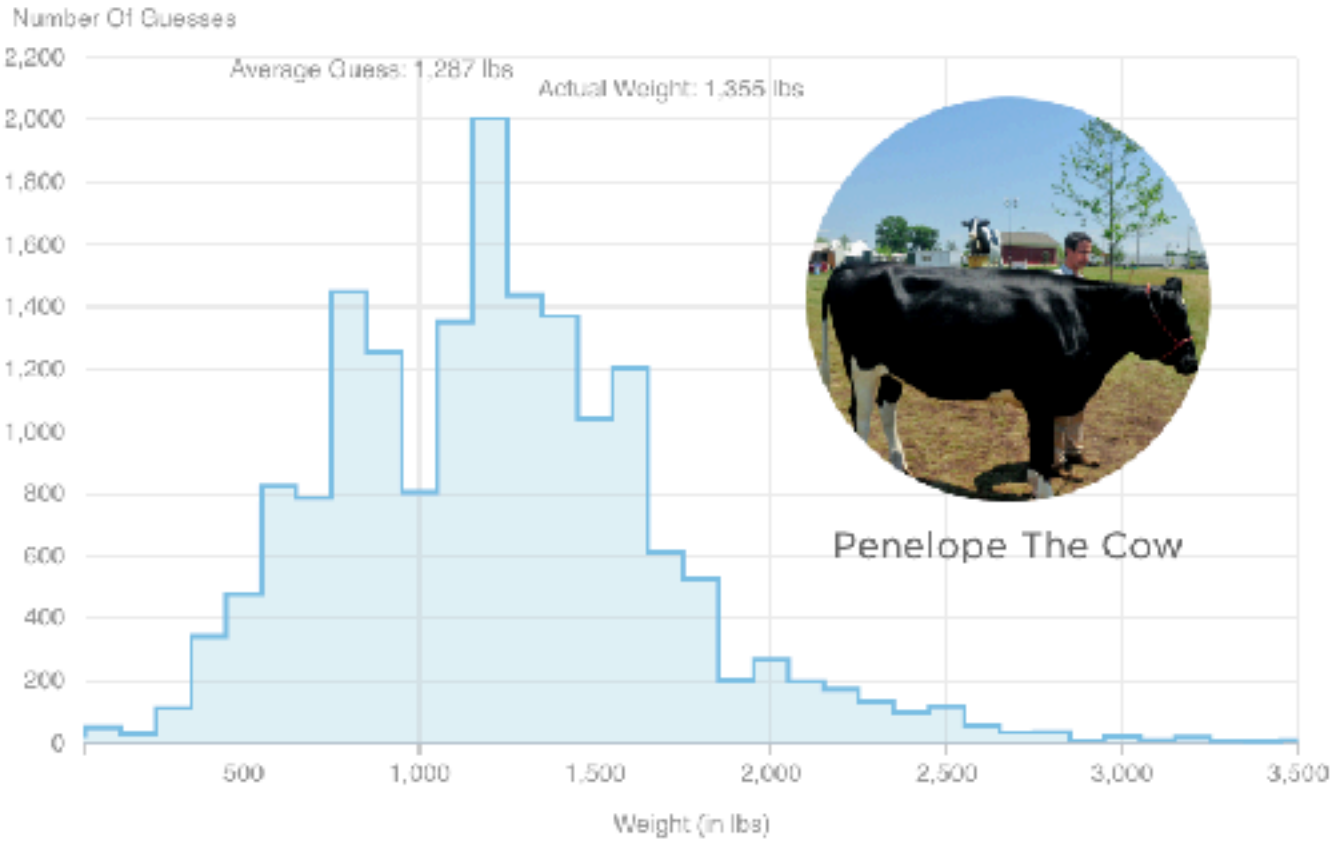
Revise Time Estimate

- Do you have comparable experience to base an estimate off of?
- 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

S3D

How Much Does This Cow Weigh?

(All People)



Source: The Internet.

Credit: Quoctrung Bui/NPR

S3D

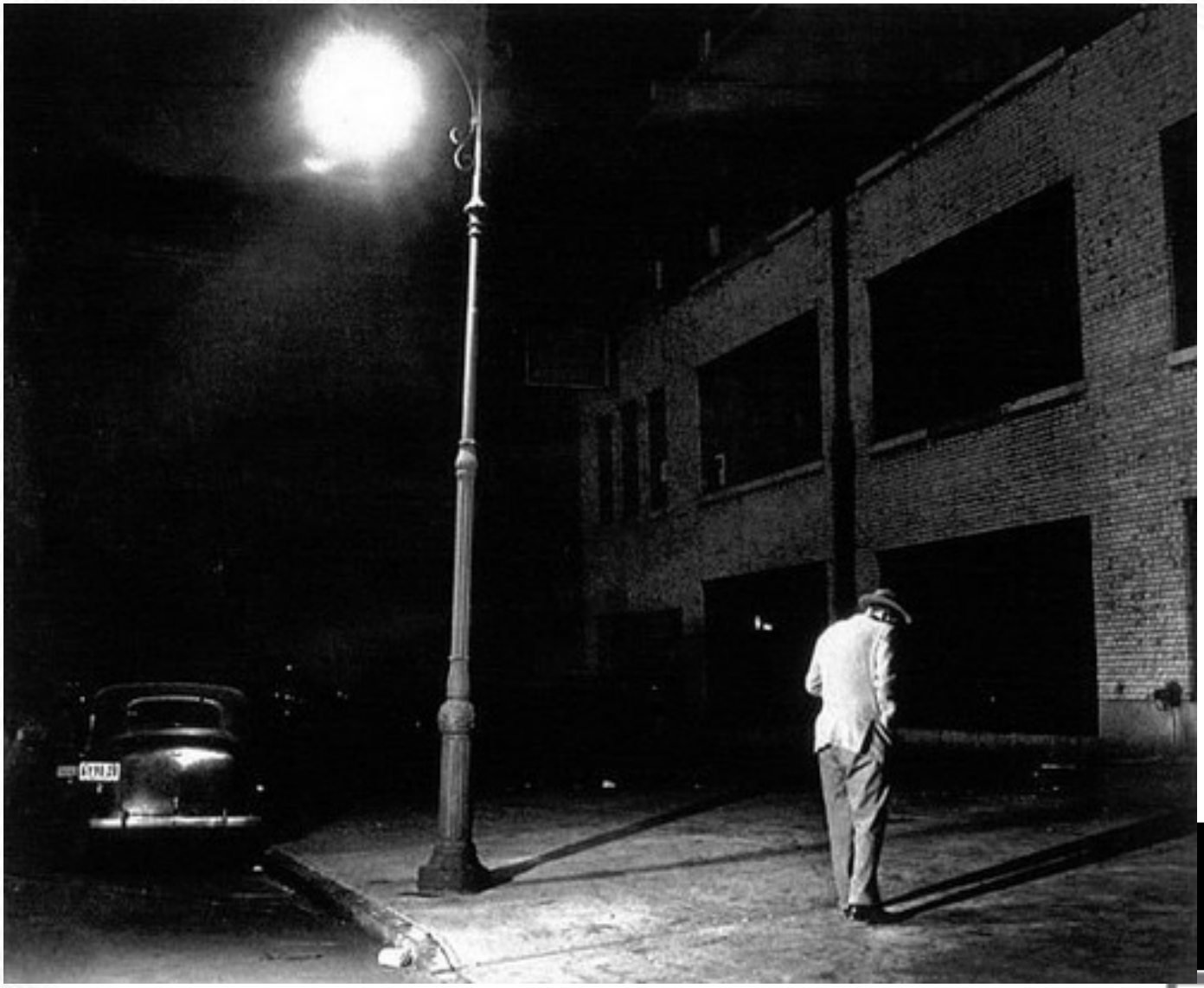

XS
S
M
L
XL
made by :codica
codica.com

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?



S3D

Milestones and deliverables make progress *observable*

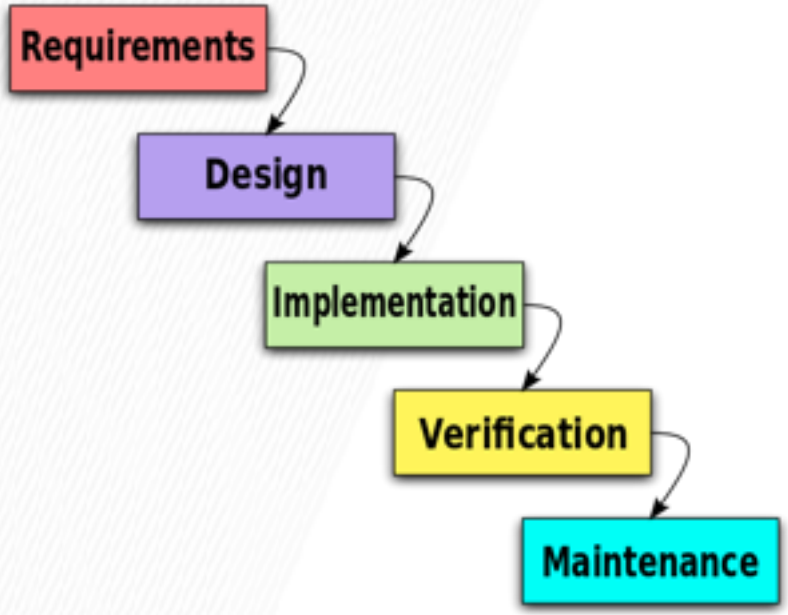
Milestone: clear end point of a (sub)tasks

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

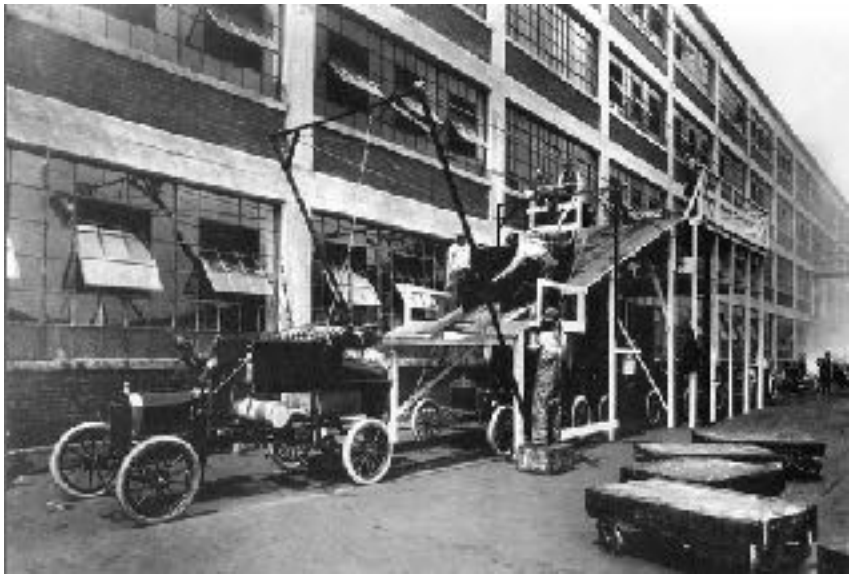
Deliverable: Result for customer

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

Waterfall model was the original software process



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



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

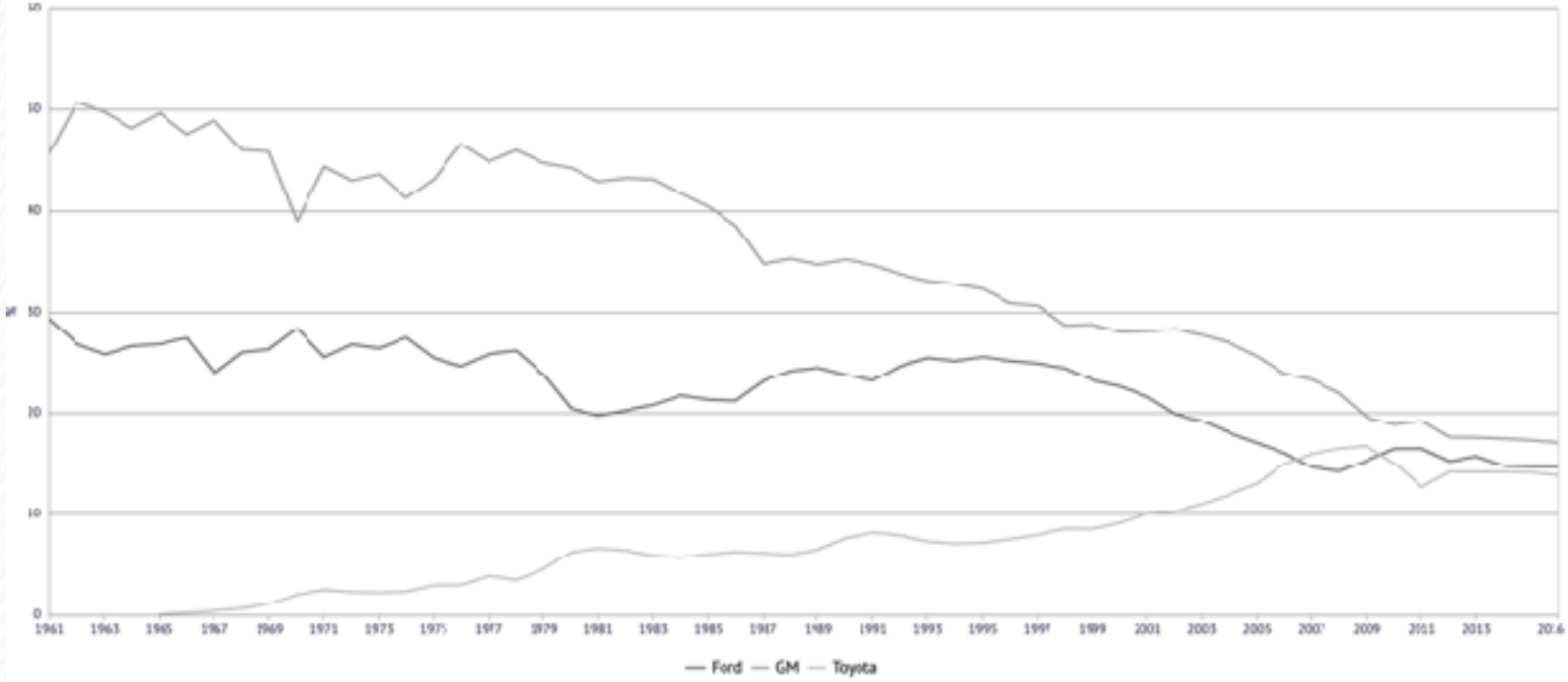
Lots of software buzzwords invented recently build on these ideas

- Just-in-time, DevOps, Shift-Left

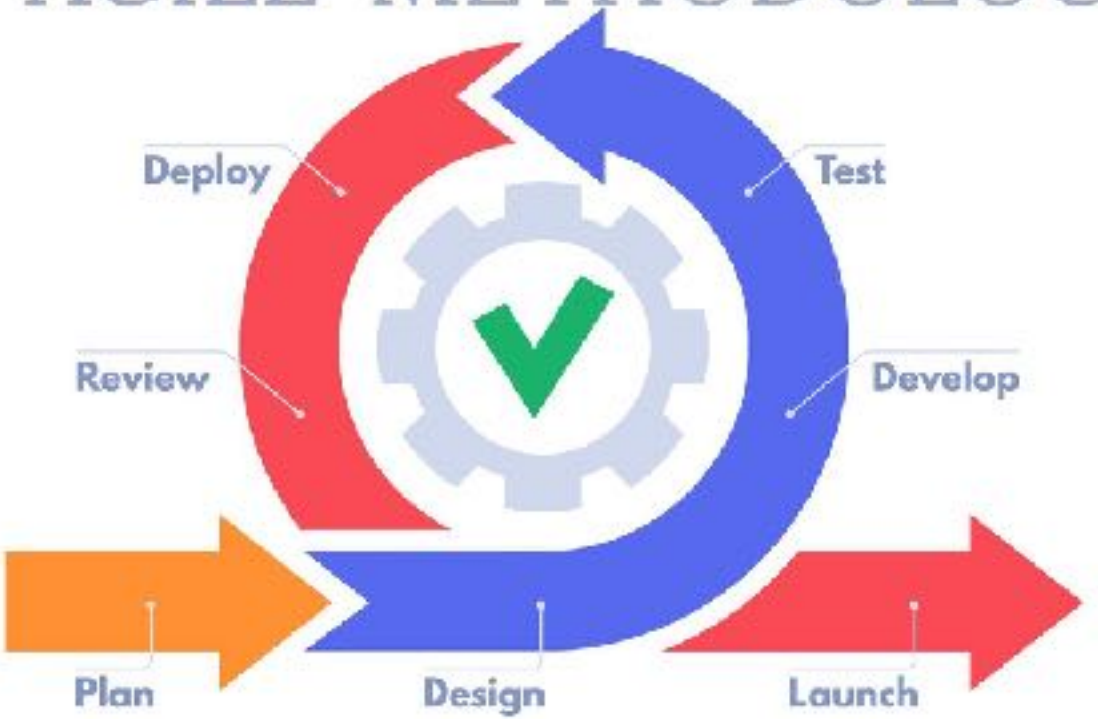
See also: “The machine that changed the world” by James P Womack et al. The Free Press, 2007.

S3D

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



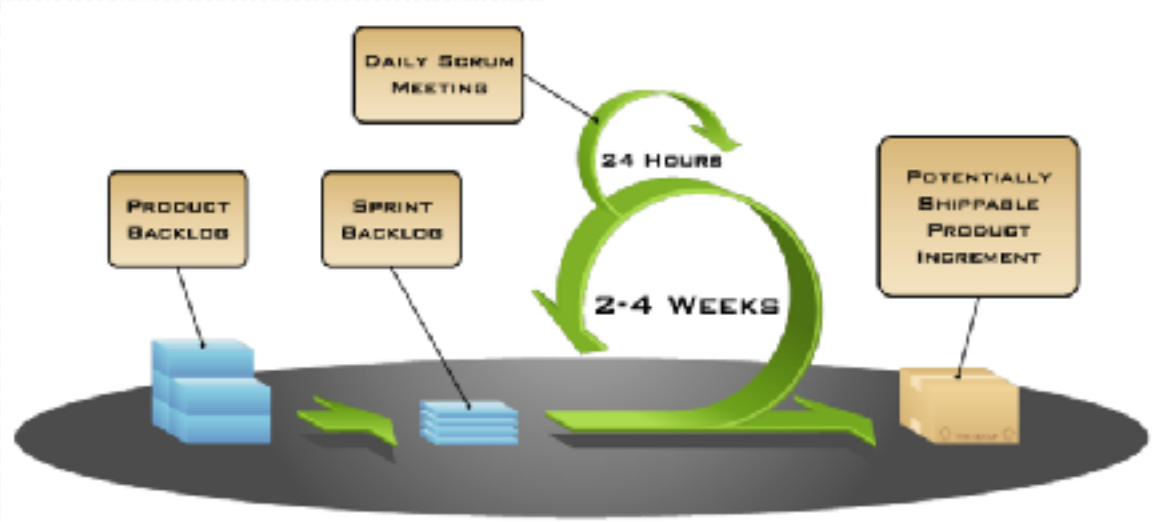
AGILE METHODOLOGY



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

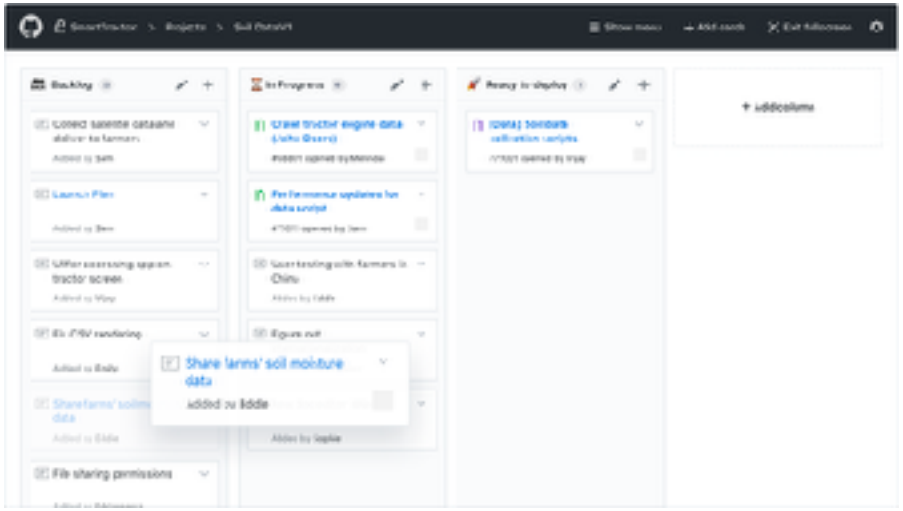
The **product backlog** is all the features for the product

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

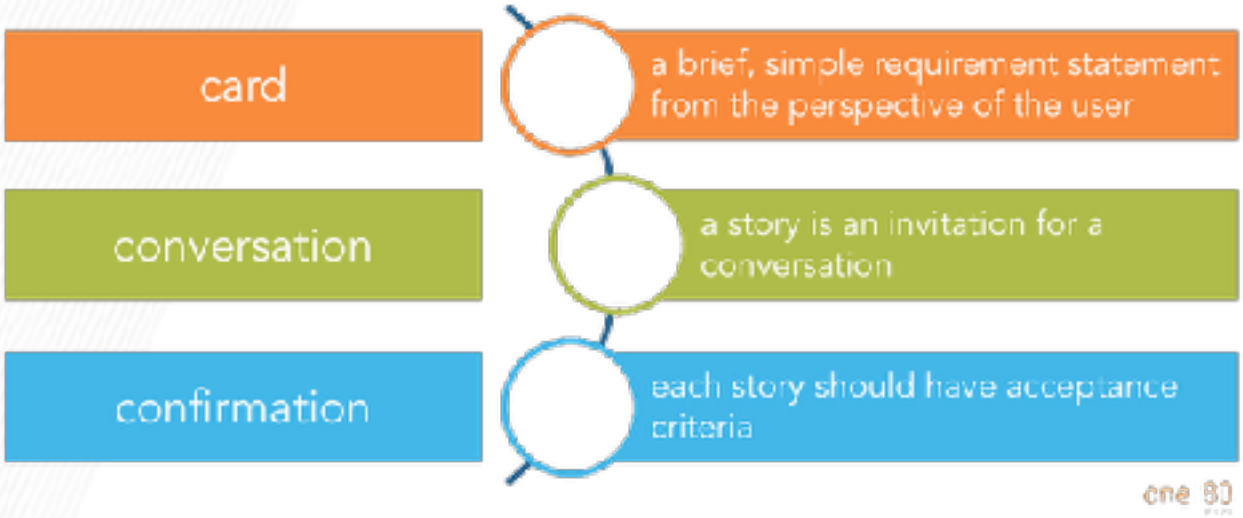
S3D

User Stories

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

S3D

User Stories



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

Fred Brooks, on requirements.

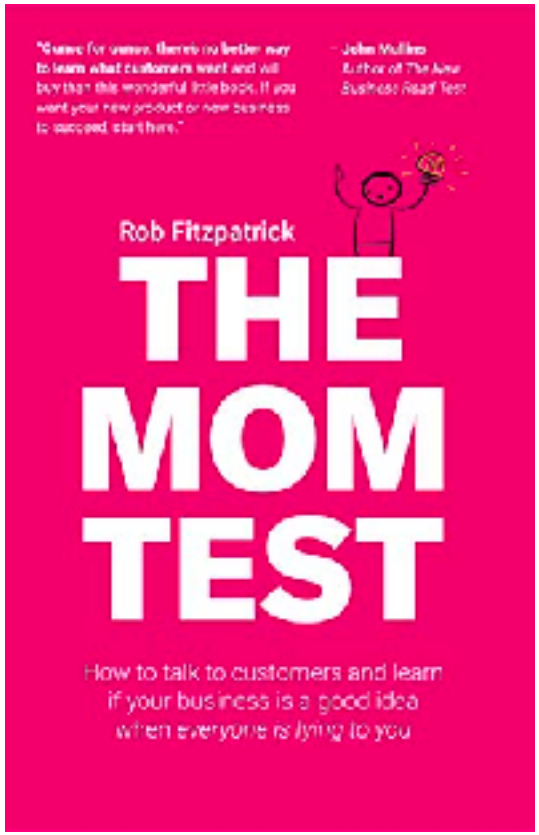
The hardest single part of building a software system is deciding precisely what to build.
No other part of the conceptual work is as difficult as establishing the detailed technical requirements …
No other part of the work so cripples the resulting system if done wrong.
No other part is as difficult to rectify later.

— Fred Brooks

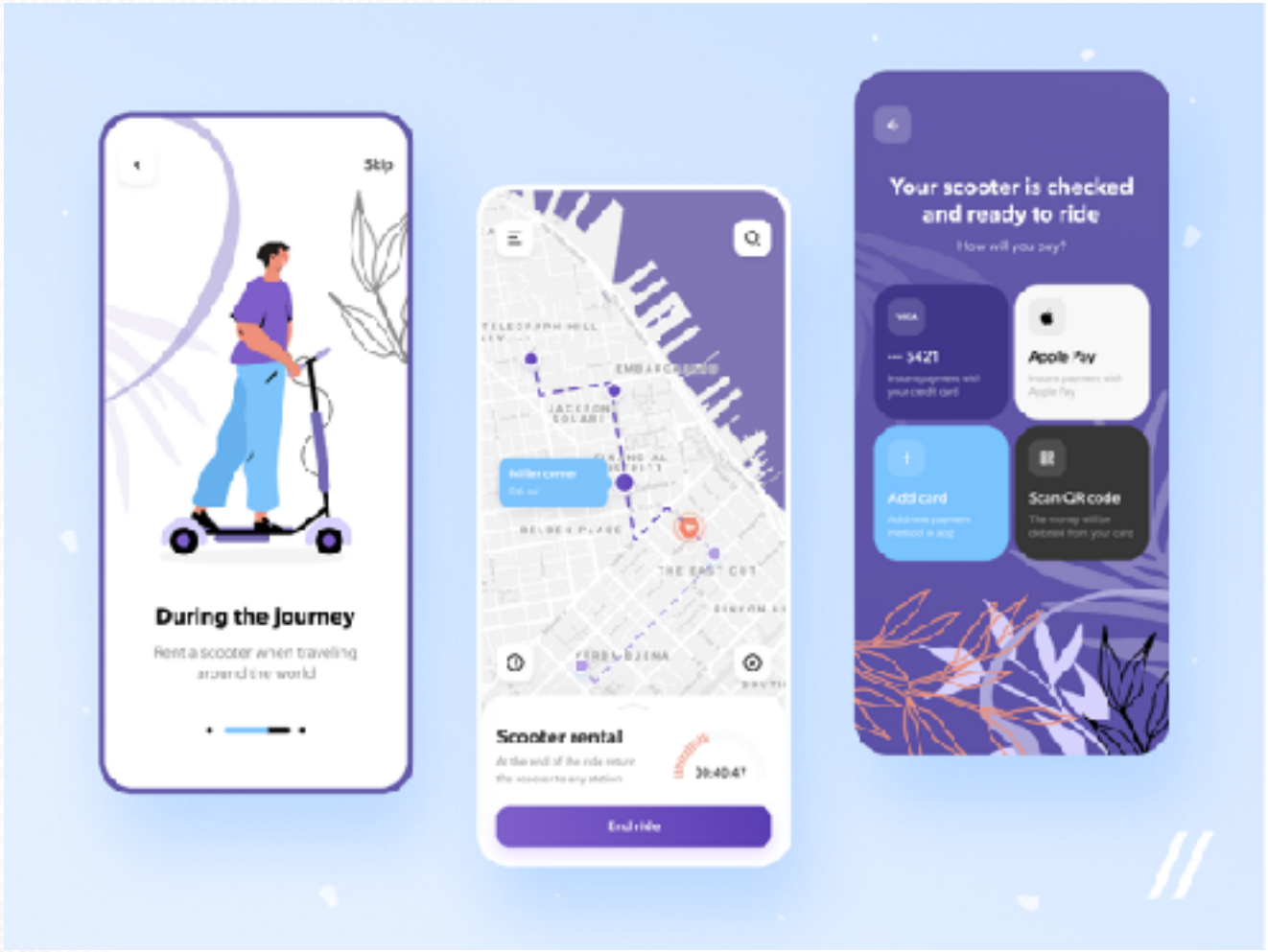
S3D

“The Mom Test”

- Talk about their life instead of your idea
- Ask about specifics in the past instead of generics or opinions about the future
- Talk less and listen more



Exercise



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

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/

S3D

Independent

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

S3D

Negotiable

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

S3D

Valuable

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

S3D

Estimable

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

S3D

Small

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

S3D

Testable

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

S3D

Activity

Follow the INVEST guidelines for good user stories!

one 80

S3D