

SEG 2105 - LECTURE 01

SOFTWARE AND SOFTWARE ENGINEERING

THE NATURE OF SOFTWARE...

ENGINEERING

tangible

[tan-juh-buhl] [SHOW IPA](#) 

[SYNONYMS](#) | [EXAMPLES](#) | [WORD ORIGIN](#)

[SEE MORE SYNONYMS FOR *tangible* ON THESAURUS.COM](#)

adjective

- 1 capable of being touched; discernible by the touch; material or substantial.
- 2 real or actual, rather than imaginary or visionary:
the tangible benefits of sunshine.
- 3 definite; not vague or elusive:
no tangible grounds for suspicion.
- 4 (of an asset) having actual physical existence, as real estate or chattels, and therefore capable of being assigned a value in monetary terms.

<https://www.dictionary.com/browse/tangible>

INTANGIBLE



Hard to understand development effort

tangible

[tan-juh-buhl] [SHOW IPA](#)

[SYNONYMS](#) | [EXAMPLES](#) | [WORD ORIGIN](#)

[SEE MORE SYNONYMS FOR *tangible* ON THESAURUS.COM](#)

adjective

- 1 capable of being touched; discernible by the touch; **material or substantial.**
- 2 **real or actual, rather than imaginary or visionary:**
the tangible benefits of sunshine.
- 3 **definite; not vague or elusive:**
no tangible grounds for suspicion.
- 4 **(of an asset) having actual physical existence, as real estate or chattels, and therefore capable of being assigned a value in monetary terms.**



Software does NOT have these qualities



<https://www.dictionary.com/browse/tangible>

SOFTWARE IS EASY TO REPRODUCE

How many Droplets?

Deploy multiple Droplets with the same configuration.

-10 Droplets+

You've reached the maximum



Do you want 1 or 100000 servers?

Choose a hostname

Give your Droplets an identifying name using alphanumeric characters, dashes, and underscores.

ubuntu-s-4vcpu-8gb-nyc1-01

ubuntu-s-4vcpu-8gb-nyc1-02

ubuntu-s-4vcpu-8gb-nyc1-03

ubuntu-s-4vcpu-8gb-nyc1-04

ubuntu-s-4vcpu-8gb-nyc1-05

Cost in development

BUT, very labour intensive, and difficult to automate

Droplets
Create cloud servers

Clusters
Create Kubernetes clusters

Databases
Create database clusters

Volumes
Add storage to Droplets

Domains/DNS
Route your existing domains

Cloud Firewalls
Increase Droplet security

Floating IPs
Reserve IP addresses for Droplets

Load Balancers
Distribute traffic to Droplets

Alert Policies
Monitor your Droplets

Spaces
Store and serve static assets

OTHER ENGINEERING PRODUCTS . . .

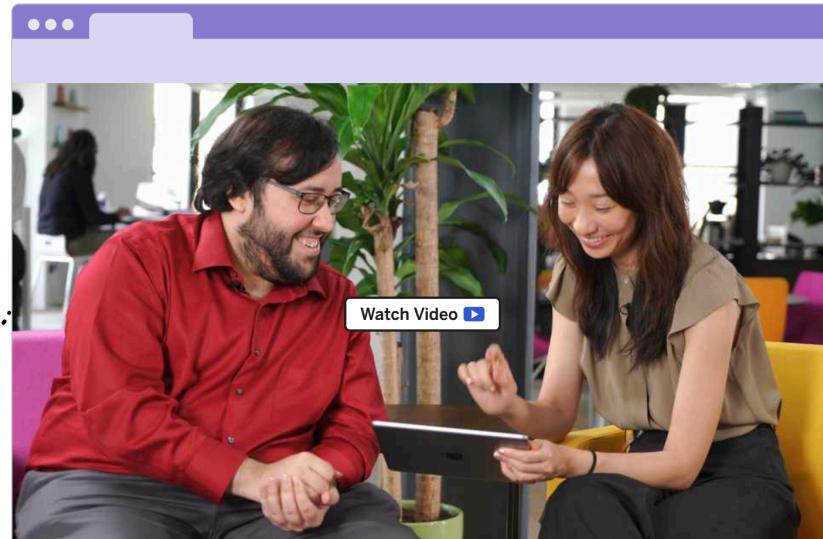


<https://www.touropia.com/most-famous-bridges-in-the-world/>

Manufacturing is
the most costly
stage (usually)



ANYONE CAN BUILD SOFTWARE



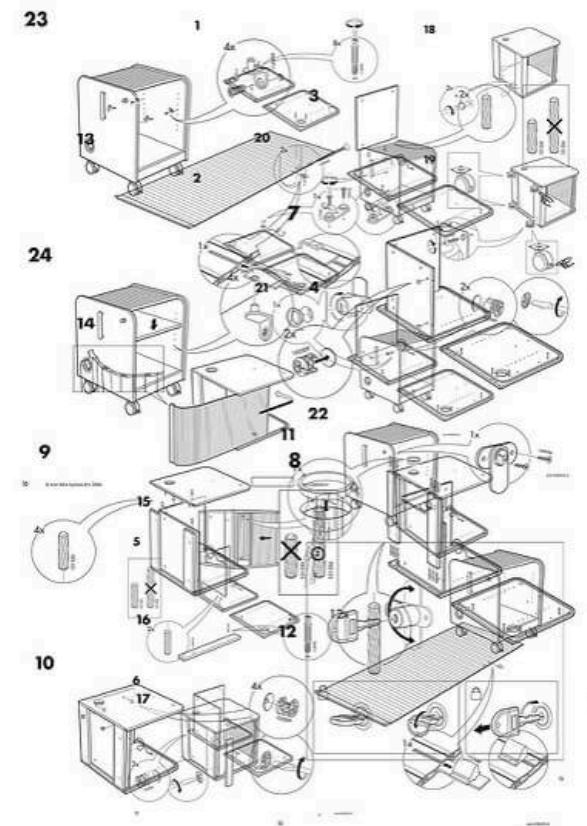
The screenshot shows the Glitch website homepage. At the top right, there is a search bar with the placeholder "bots, apps, users", a magnifying glass icon, a "New Project" button, and a "Sign in" button. Below the search bar is a large purple callout box containing the text: "Glitch is the friendly community where everyone builds the web". Below this text is a smaller message: "Simple, powerful, free tools to create and use millions of apps." A green button labeled "Start Creating →" is at the bottom of the callout. To the right of the callout, a man with glasses and a beard, wearing a red shirt, and a woman with long dark hair, wearing a tan top, are sitting on a couch, looking at a laptop together. A "Watch Video" button with a play icon is overlaid on the video frame. At the bottom of the page are three colorful callout boxes: an orange one on the left with icons for a cat, a diamond, and a folder, and the text "Spark your next project"; a purple one in the middle with a brain icon and a smartphone, and the text "Just start typing"; and a blue one on the right with a code editor icon and a smiley face, and the text "Code together".

<https://glitch.com/>

UNTRAINED PEOPLE CAN HACK SOMETHING TOGETHER



<https://www.youtube.com/watch?v=wXjs6DXoeY&feature=youtu.be>



SOFTWARE DOES NOT 'WEAR OUT'

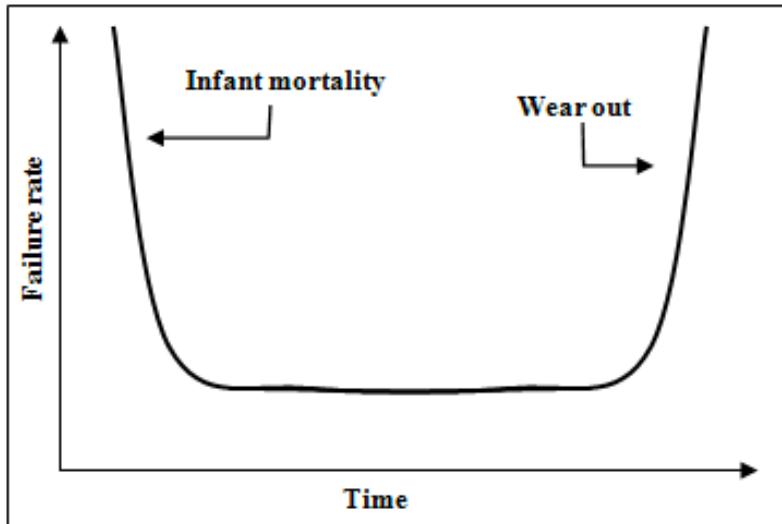


Figure 01: Failure curve for hardware

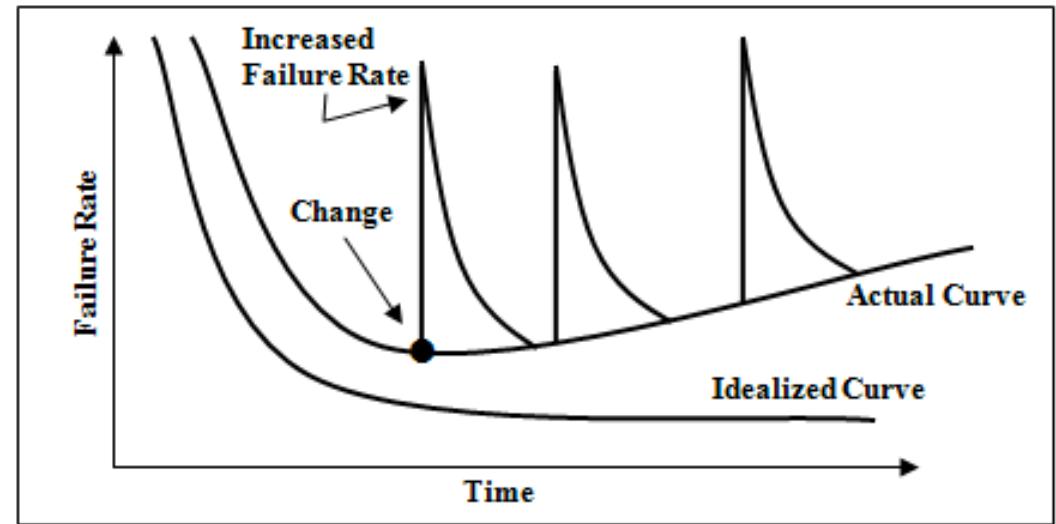


Figure 02: Failure Curves for software

<https://www.onlineclassnotes.com/2013/01/software-doesnt-wear-out-explain-this.html>

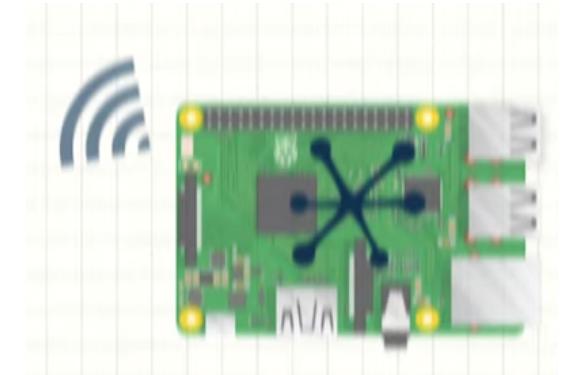
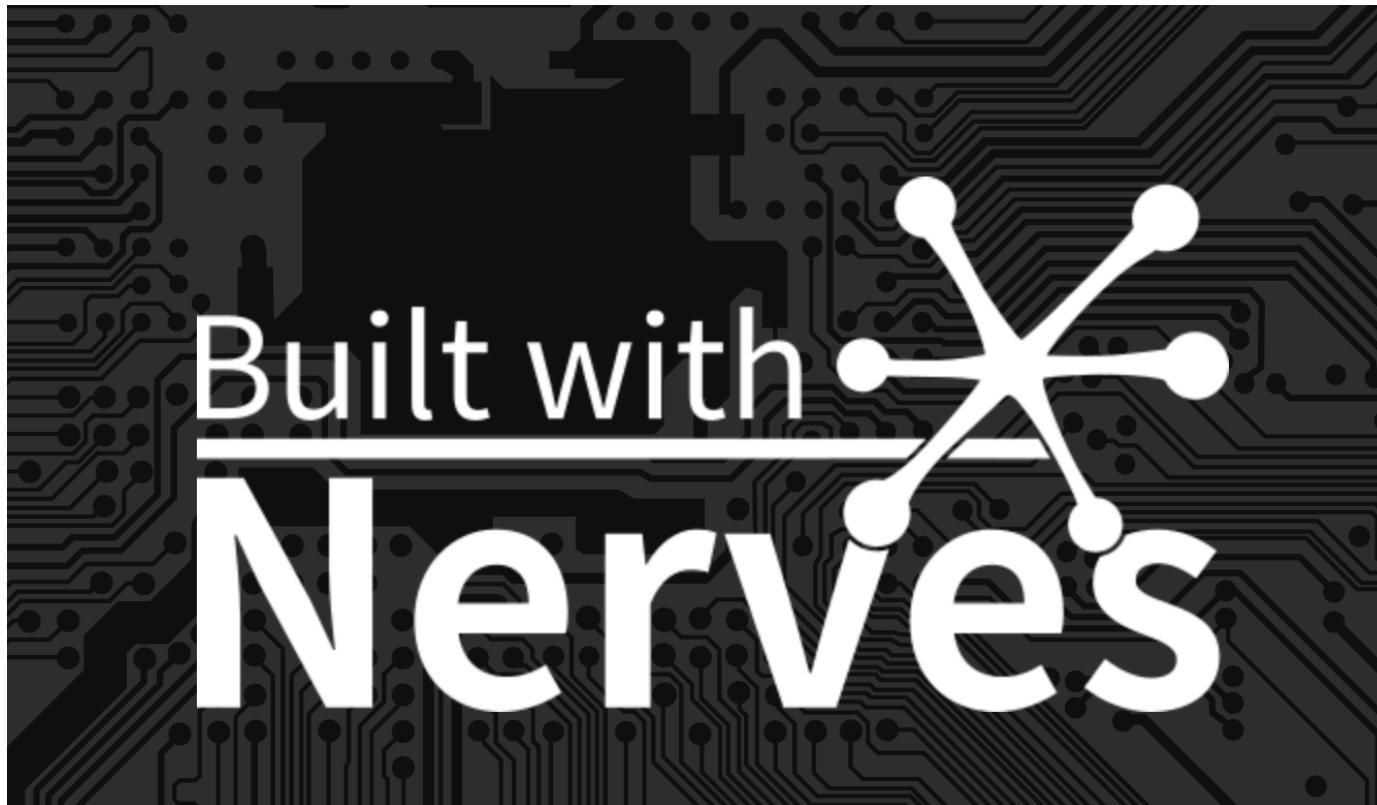
MUCH SOFTWARE HAS POOR DESIGN
AND IS GETTING WORSE

**WE HAVE TO LEARN TO 'ENGINEER'
SOFTWARE**

WHAT KINDS OF S/W DO WE BUILD

SOFTWARE TAXONOMY

REAL TIME EMBEDDED SOFTWARE

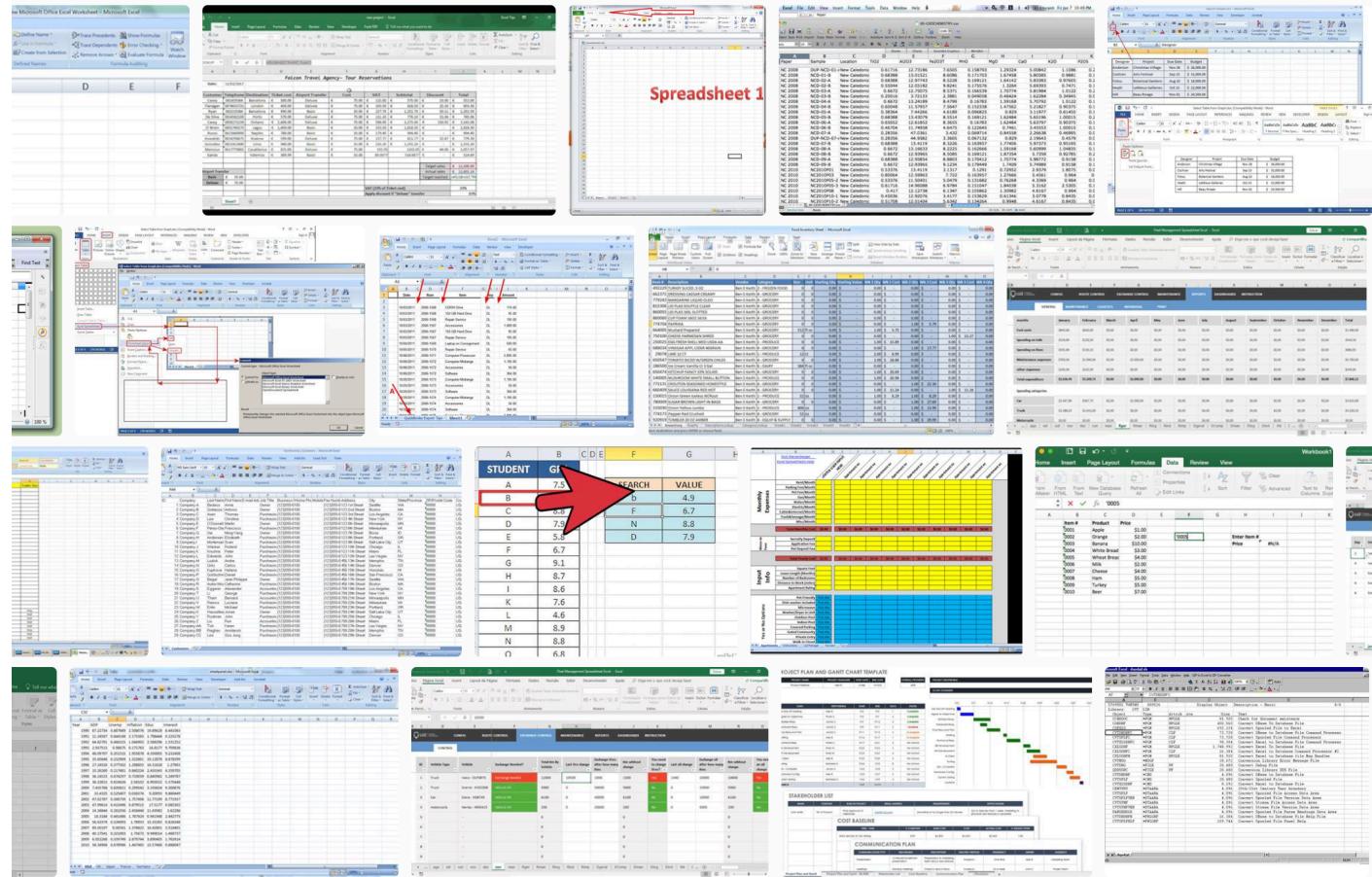


E.g. control and monitoring systems

Must react immediately

Safety often a concern

DATA PROCESSING SOFTWARE



Runs your business

Accuracy and security of data are key

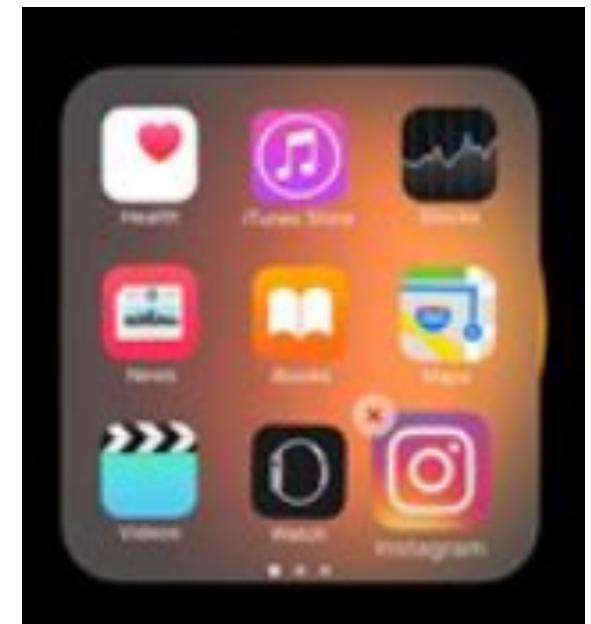


Games

Web-based
software

Mobile Apps

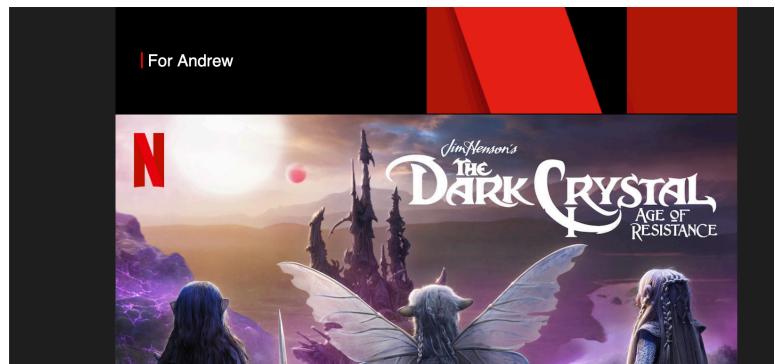
Machine
Learning



Andrew, we just added a TV show you might like ➔

Netflix <info@mailer.netflix.com> [Unsubscribe](#)
to me ▾

Fri, Aug 30, 4:48 PM (22 hours ago) ⭐



SOFTWARE TAXONOMY FOR ESTIMATING PROJECT SIZE

Scope:	Class:	Type:
1) all that needs to be written is a function. 2) module 3) reusable module 4) disposable prototype 5) evolutionary prototype 6) standalone program 7) component of a system 8) release of system 9) new system 10) compound system	1) individual software 2) shareware 3) academic software 4) single location – internal 5) multi location – internal 6) contract project – civilian 7) time sharing system 8) military services 9) internet 10) leased software 11) bundled software 12) marketed commercially 13) outsourced contract 14) government contract 15) military contract	1) nonprocedural 2) web applet 3) batch (not database) 4) interactive 5) interactive GUI 6) batch database 7) interactive database 8) client/server 9) mathematical 10) systems 11) communications 12) process control 13) trusted system 14) embedded 15) image processing 16) multimedia 17) robotics 18) artificial intelligence 19) neural net 20) hybrid: mixed

SOFTWARE TAXONOMY FOR ESTIMATING PROJECT SIZE

Scope:	Class:	Type:
1) all that needs to be written is a function.	1) individual software	1) nonprocedural
2) module	2) shareware	2) web applet
3) reusable module	3) academic software	3) batch (not database)
4) disposable prototype	4) single location – internal	4) interactive
5) evolutionary prototype	5) multi location – internal	5) interactive GUI
6) standalone program	6) contract project – civilian	6) batch database
7) component of a system	7) time sharing system	7) interactive database
8) release of system	8) military services	8) client/server
9) new system	9) internet	9) mathematical
10) compound system	10) leased software	10) systems
	11) bundled software	11) communications
	12) marketed commercially	12) process control
	13) outsourced contract	13) trusted system
	14) government contract	14) embedded
	15) military contract	15) image processing
		16) multimedia
		17) robotics
		18) artificial intelligence
		19) neural net
		20) hybrid: mixed

Jones Very Early Size Predictor



ANOTHER TAXONOMY

A Data-dominant software

- A.con Consumer-oriented software
- A.bus Business-oriented software
- A.des Design and engineering software
- A.inf Information display and transaction entry

B. Systems software

- B.os Operating systems
- B.net Networking / Communications
- B.dev Device / Peripheral drivers
- B.ut Support utilities
- B.mid Middleware and system components
- B.bp Software Backplanes (e.g. Eclipse)
- B.svr. Servers
- B.mal Malware

C. Control-dominant software

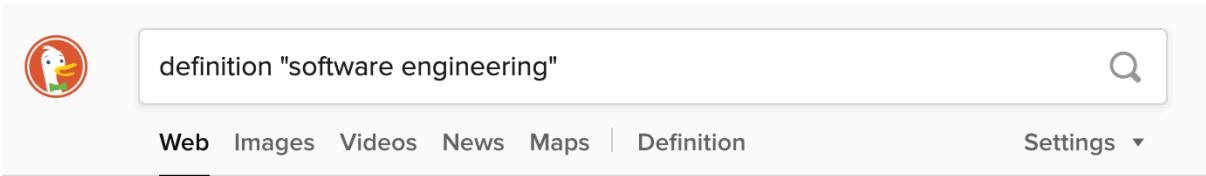
- C.hw. Hardware control
- C.em. Embedded software
- C.rt. Real time control software
- C.pc. Process control software (i.e. air traffic control, industrial process, nuclear plants)

D. Computation-dominant software

- D.or. Operations research
- D.im. Information management and manipulation
- D.art. Artistic creativity
- D.sci Scientific software
- D.ai Artificial intelligence

WHAT IS

SOFTWARE ENGINEERING



A screenshot of a search engine interface. The search bar at the top contains the query "definition \"software engineering\"". Below the search bar are navigation links: Web, Images, Videos, News, Maps, Definition (which is underlined), and Settings. There are also filters for location ("Canada"), safe search ("Moderate"), and time ("Any Time").

What is Software Engineering? - Definition from Techopedia

 <https://www.techopedia.com/definition/13296/software-engineering>

Software engineering is the process of analyzing user needs and designing, constructing, and testing end user applications that will satisfy these needs through the use of software programming languages. It is the application of engineering principles to software development.

Software Engineering | Definition of Software Engineering by ...

 [https://www.merriam-webster.com/dictionary/software engineering](https://www.merriam-webster.com/dictionary/software%20engineering)

Software engineering definition is - a branch of computer science that deals with the design, implementation, and maintenance of complex computer programs.

What is Software Engineering? Definition of Software ...

 <https://economictimes.indiatimes.com/definition/software-engineering>

Software engineering is a detailed study of engineering to the design, development and maintenance of software. Software engineering was introduced to address the issues of low-quality software projects. Problems arise when a software generally exceeds timelines, budgets, and reduced levels of quality. It ensures that the application is built ...

Software-engineering dictionary definition | software ...

 <https://www.yourdictionary.com/software-engineering>

software-engineering definition: Software engineering is the process of making, testing and documenting computer programs. (noun) An example of software engineering is making a new word processing program....

The process of **solving** customers' problems by the **systematic** development and evolution of large, high-quality software systems **within cost, time and other constraints**

*That's not to say that the others [coder, developers, etc] are thought of as reckless, just that a **Software Engineer** has the connotation of **not being reckless**.*

– Daniel Kaplan (2014)

Scientists investigate that which already is; engineers create that which as never been.

– Albert Einstein (1879 - 1955)

SOLVING CUSTOMERS' PROBLEMS





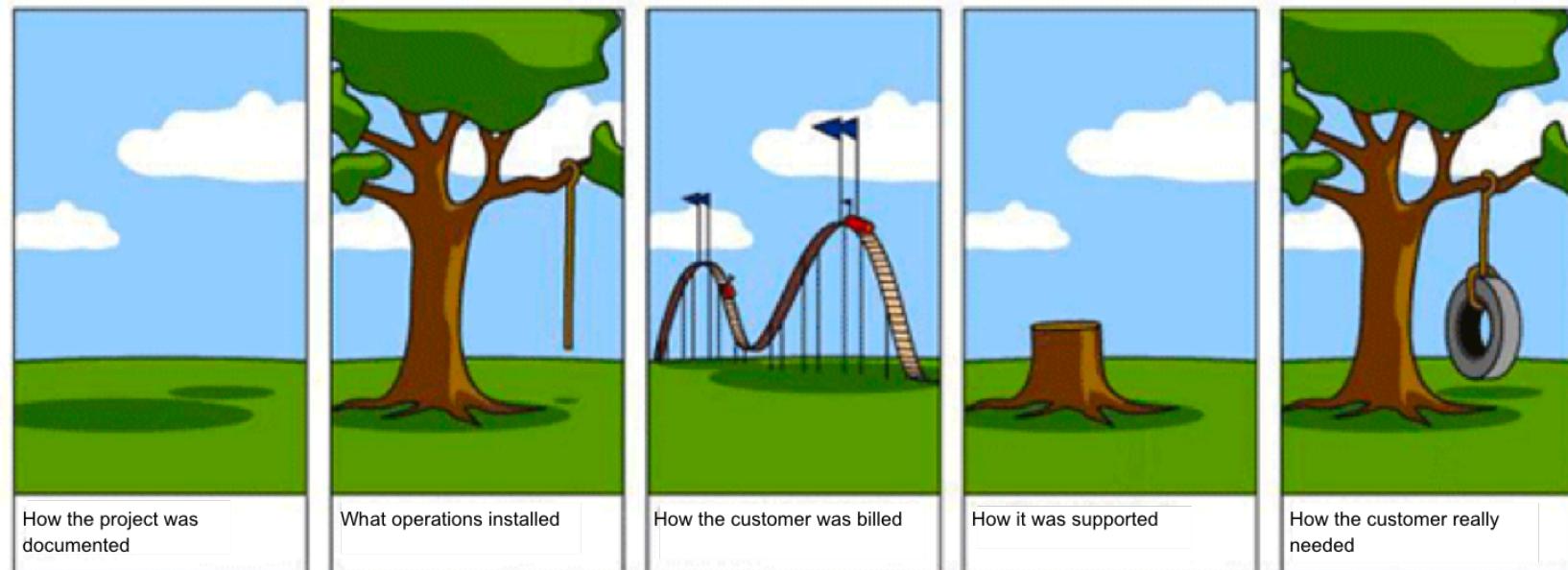
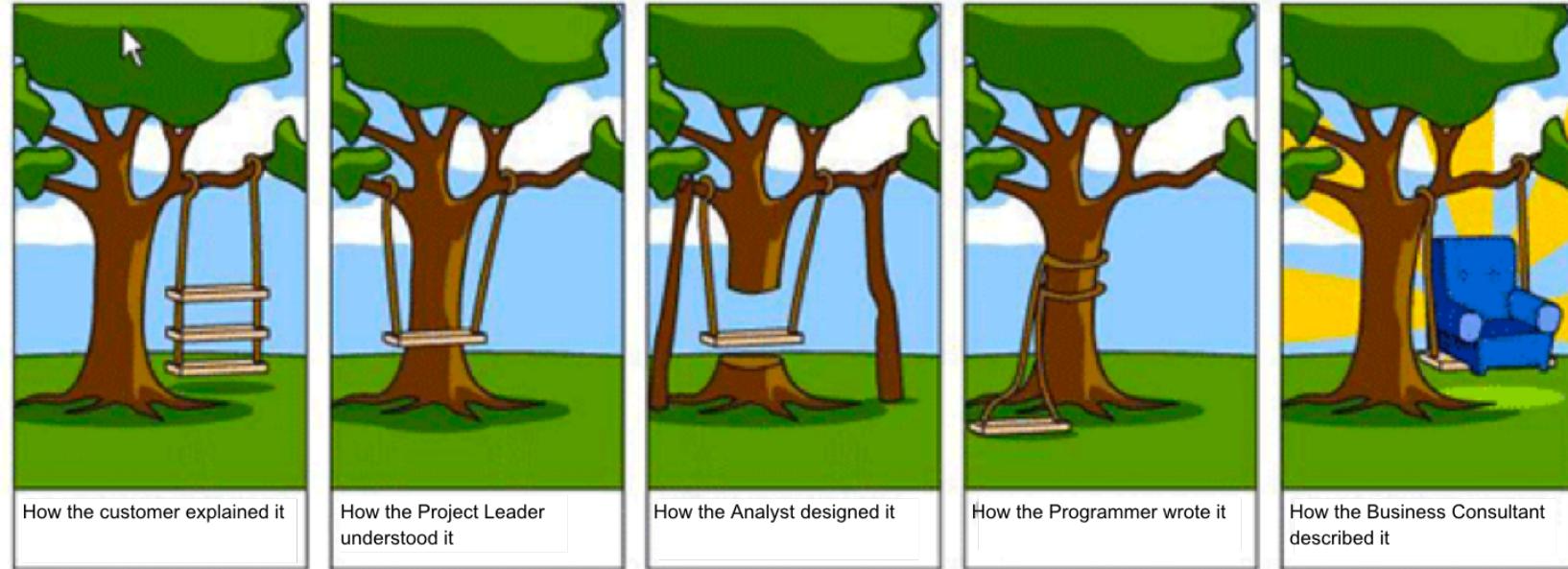
<https://blog.professorbeekums.com/build-vs-buy-decisions/>

ADDING UNNECESSARY FEATURES OFTEN MAKES SOFTWARE WORSE



<http://vignette4.wikia.nocookie.net/simpsons/images/0/05/TheHomer.png>

SOFTWARE ENGINEERS MUST COMMUNICATE EFFECTIVELY TO IDENTIFY AND UNDERSTAND THE PROBLEM



Most systematic
development work is
evolution



An **engineering** process involves
applying well understood **techniques**
in a **organized** and **disciplined** way



Many well-accepted
practices have been
formally standardized



e.g IEEE or ISO

IEEE.org IEEE Xplore Digital Library IEEE Standards IEEE Spectrum More Sites eTools Sign in 

IEEE STANDARDS ASSOCIATION Standards Products & Services Technologies & Initiatives Participate Software I  MAC ADDRESS BUY STANDARDS

Showing results for: **Software+Engineering** Change

Software+Engineering 

Narrow your search results by combining terms with a plus sign (e.g. Ethernet+Power)

Content Type ▾

- All
- Event
- Events
- Governance
- Initiative
- News
- Products
- Project
- Services

+ SHOW MORE reset

763 results found. Results are sorted by relevance ◀ 1-25 ▶

IEEE 12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes
...12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes

P24748-3 - Systems and Software Engineering-Life Cycle Management-Part 3: Guidelines for the Application of ISO/IEC/IEEE 12207 (Software Life Cycle Processes)
P24748-3 - Systems and Software Engineering-Life Cycle Management-Part 3: Guidelines for the Application...

IEEE 15026-1-2011 - IEEE Trial-Use Standard--Adoption of ISO/IEC TR 15026-1:2010 Systems and Software Engineering--Systems and Software Assurance--Part 1: Concepts and Vocabulary
...TR 15026-1:2010 Systems and Software Engineering--Systems and Software Assurance--Part 1: Concepts...

<https://standards.ieee.org/search-results.html?q=Software+Engineering>



- Standards
- All about ISO
- Taking part
- Store
- 
- 
- EN 

FILTER

 **ALL RESULTS**

 **STANDARDS (217)**

 **PAGES (2)**

 **NEWS (6)**

 **DOCUMENTS (737)**

LOOKING FOR THE FINER DETAILS?

Customize your search by combining multiple criteria

[Advanced search for standards »](#)

527 RESULTS FOUND (16 MS)

 **ISO/IEC 25051:2014 SOFTWARE ENGINEERING -- SYSTEMS AND SOFTWARE QUALITY REQUIREMENTS AND EVALUATION (SQUARE) -- REQUIREMENTS FOR QUALITY OF READY TO USE SOFTWARE PRODUCT (RUSP) AND INSTRUCTIONS FOR TESTING**

ISO/IEC 25051:2014 establishes: quality requirements for Ready to Use **Software** Product (RUSP); requirements for test documentation for the testing of RUSP, including test plan, test description, and test results; instructions for conformity evaluation of RUSP. It includes also recommendations for ...

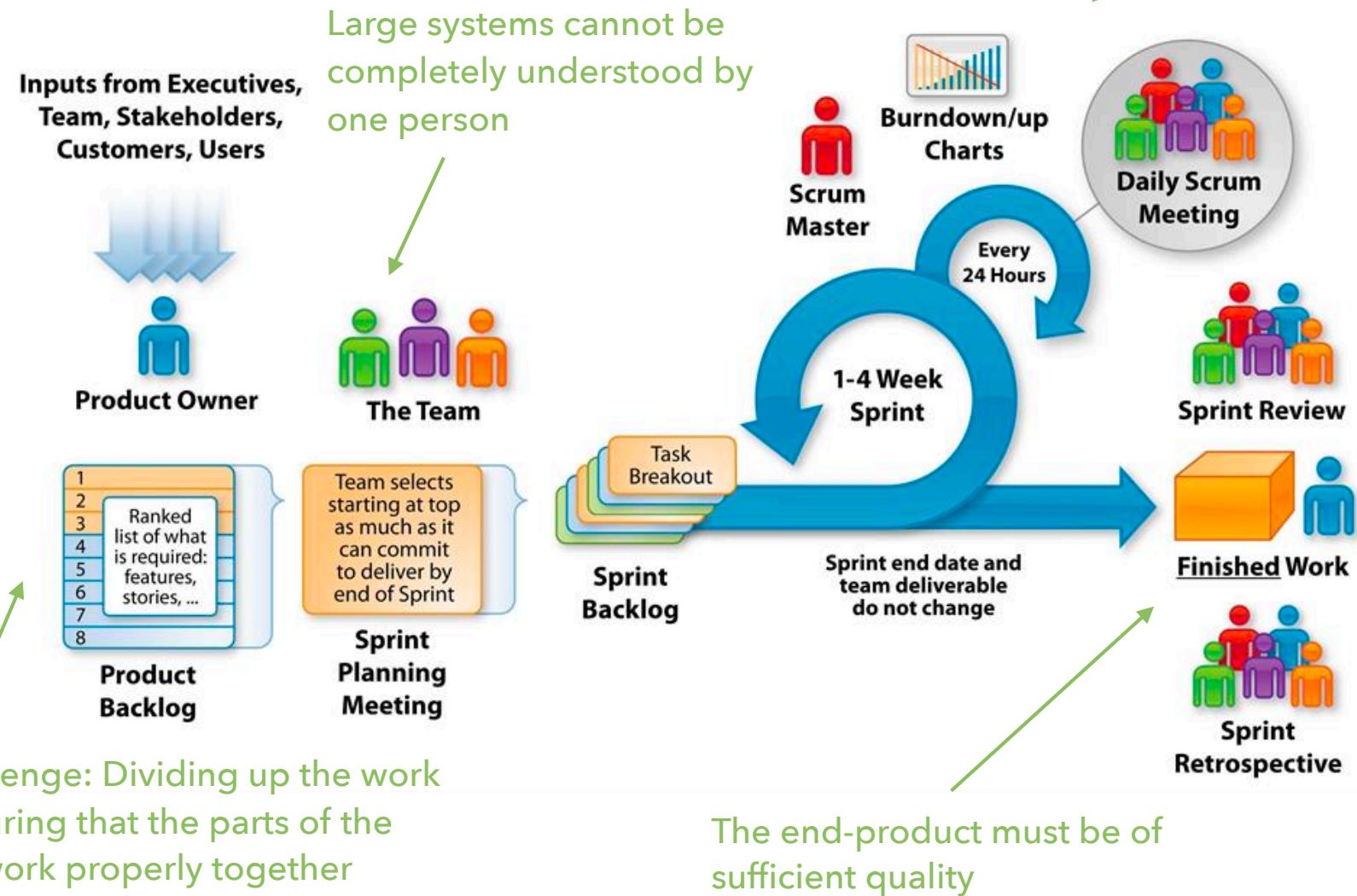
 **ISO/IEC 19761:2011 SOFTWARE ENGINEERING -- COSMIC: A FUNCTIONAL SIZE MEASUREMENT METHOD**

ISO/IEC 19761:2011 specifies the set of definitions, conventions and activities of the COSMIC Functional Size Measurement Method. It is applicable to **software** from the following functional domains: application **software**; real-time **software**; hybrids of the above. ...

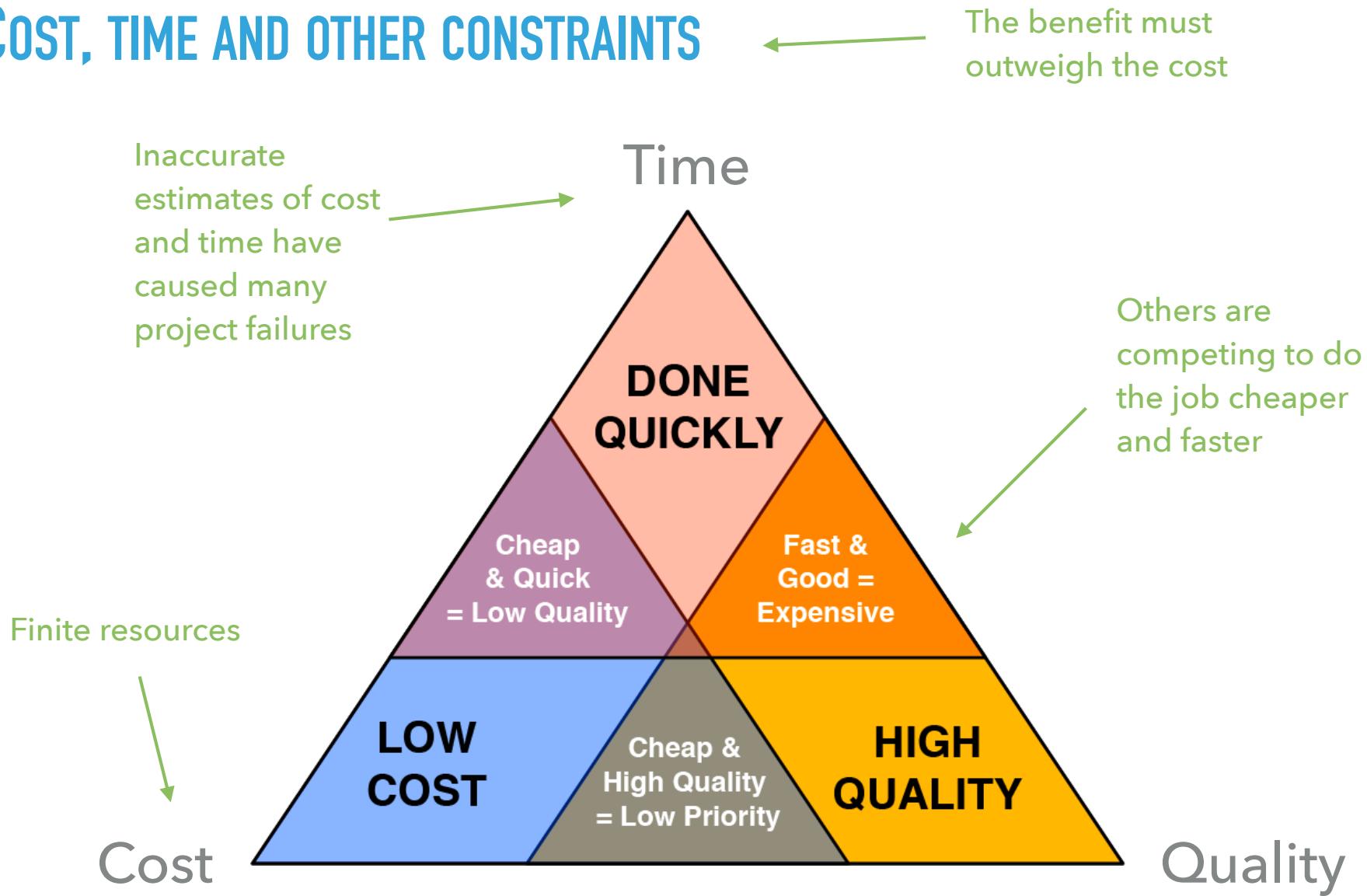
 **ISO/IEC 25062:2006 SOFTWARE ENGINEERING -- SOFTWARE PRODUCT QUALITY REQUIREMENTS AND EVALUATION (SQUARE) -- COMMON INDUSTRY FORMAT (CIF) FOR USABILITY TEST REPORTS**

ISO/IEC 25062:2006 provides a standard method for reporting usability test findings. The format is designed for reporting results of formal usability tests in which quantitative measurements were collected, and is particularly appropriate for summative/comparative testing. The CIF does not indicate how to perform ...

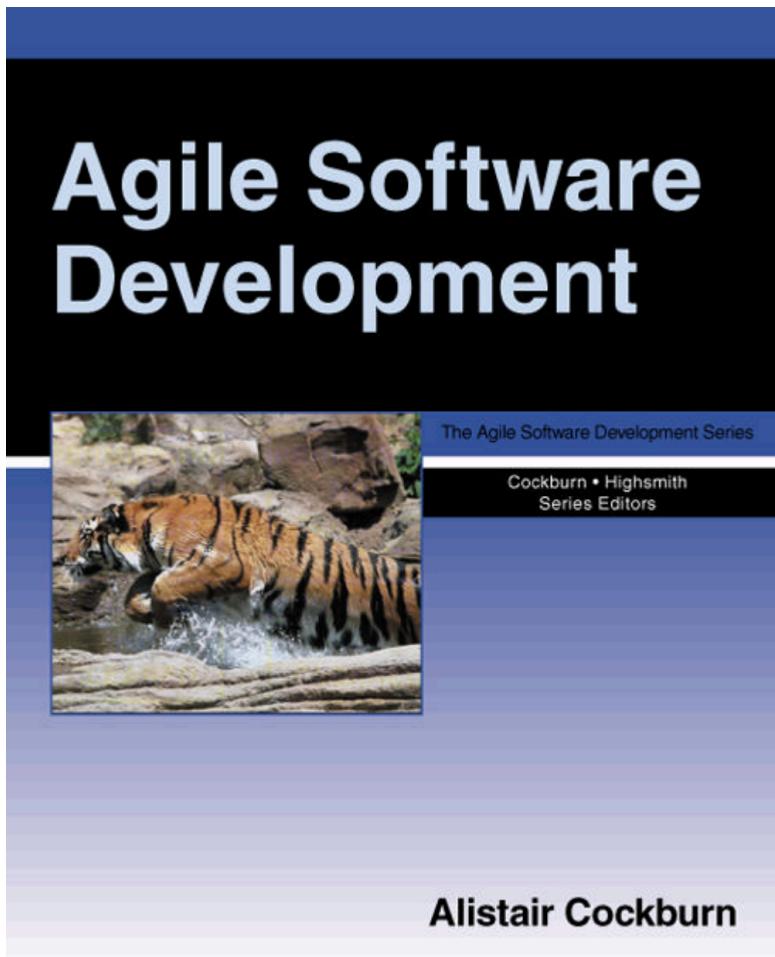
LARGE, HIGH QUALITY SOFTWARE SYSTEMS



COST, TIME AND OTHER CONSTRAINTS



SOFTWARE ENGINEERING OBJECTIVES



It considers the primary goal of the game—**delivering working software**—and the secondary goal—or residue of the game—**setting up for the next game**. The next game is altering or replacing the system, or creating a neighboring system

THE PROFESSION OF

SOFTWARE ENGINEERING

SOFTWARE CRISIS (DEFINED)

- ▶ Software crisis is a term used in the early days of computing science (first NATO Software Engineering Conference in 1968) for the difficulty of writing useful and efficient computer programs in the required time.
- ▶ The software crisis was due to the rapid increases in computer power and the complexity of the problems that could not be tackled. With the increase in the complexity of the software, many software problems arose because existing methods were insufficient.

The major cause of the software crisis is that the **machines** have become several orders of magnitude **more powerful!** To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few **weak computers, programming became a mild problem**, and now we have **gigantic computers**, programming has become an equally **gigantic problem**.

- ▶ Edsger Dijkstra, The Humble Programmer (EWD340), Communications of the ACM

THE TERM SOFTWARE ENGINEERING WAS COINED IN 1968

- ▶ People began to realize that the principles of engineering should be applied to software development
- ▶ Engineering is a licensed profession
 - ▶ In order to protect the public
 - ▶ Engineers design artifacts following well accepted practices which involve the application of science, mathematics and economics
 - ▶ Ethical practice is also a key tenet of the profession
- ▶ In many countries, much software engineering does not require an engineering license, but is still engineering

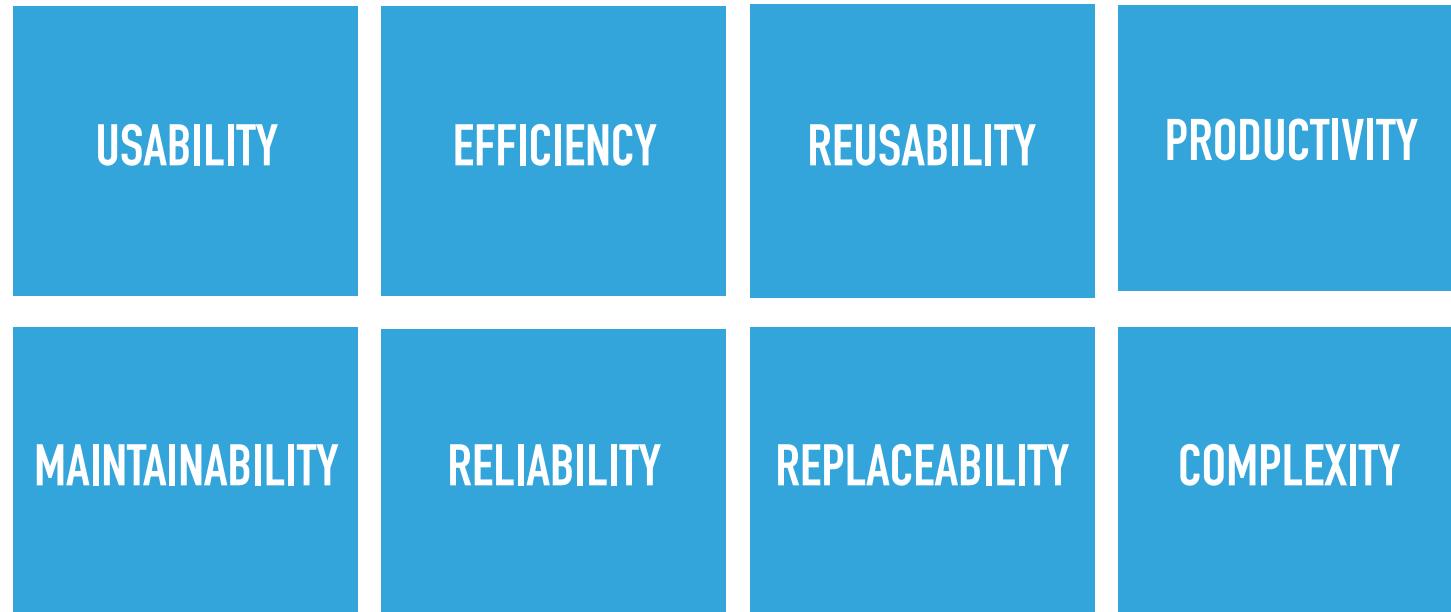
SOFTWARE ENGINEERING CODE OF ETHICS, WE SHALL . . .

- ▶ Act consistently with **public interest**
- ▶ Act in the **best interests of their clients**
- ▶ Develop and maintain with the **highest standards** possible
- ▶ Maintain **integrity and independence**
- ▶ Promote an **ethical approach in management**
- ▶ Advance the integrity and **reputation of the profession**
- ▶ Be fair and **supportive to colleagues**
- ▶ Participate in **lifelong learning**

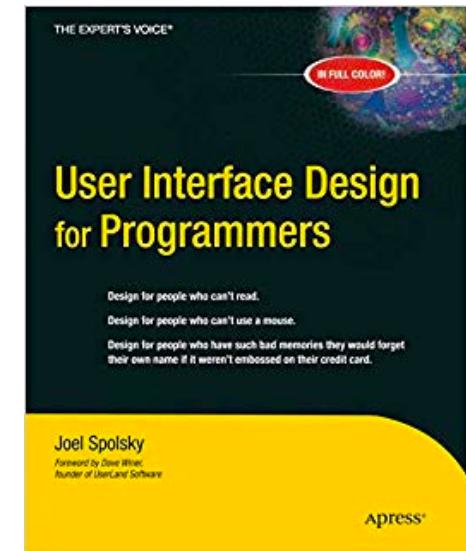
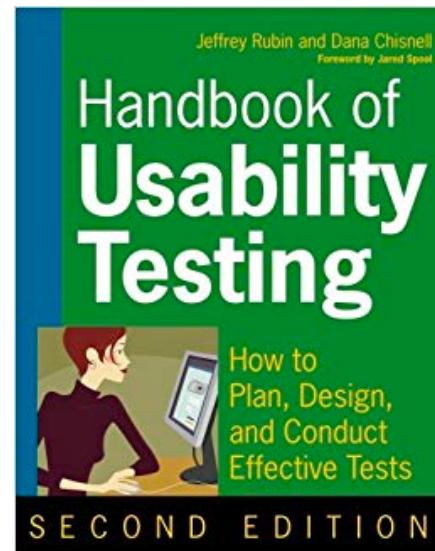
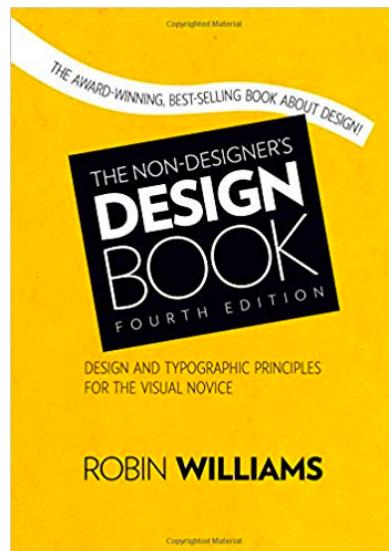
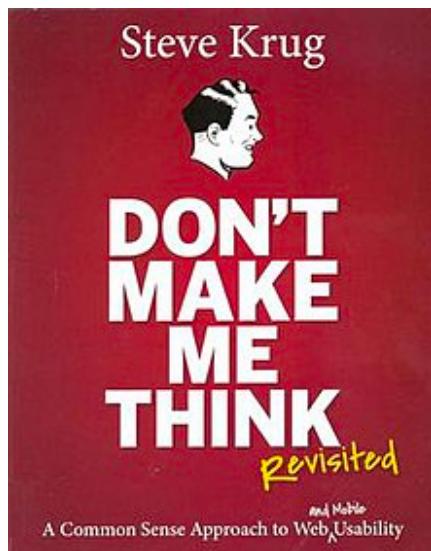
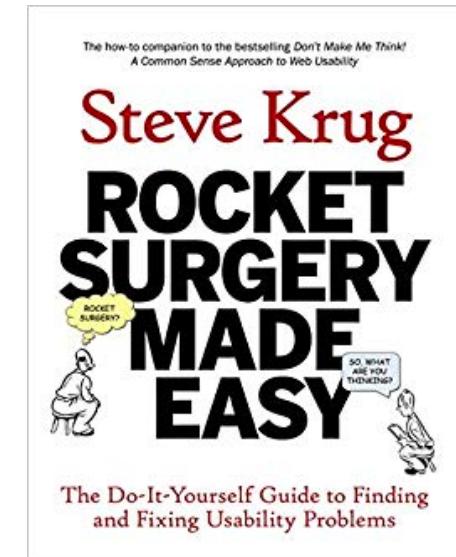
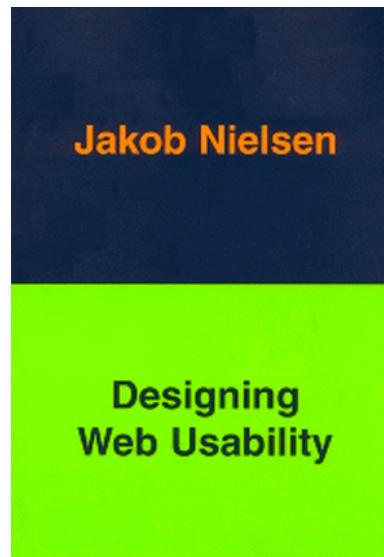
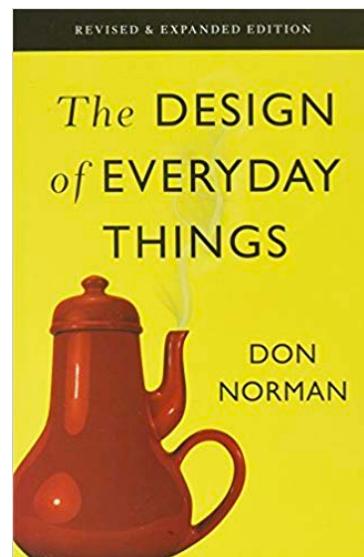
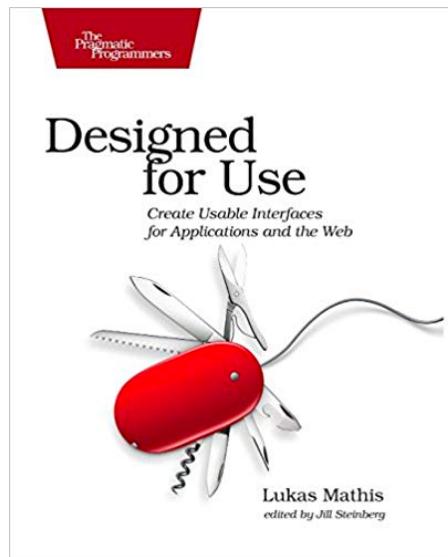
DEFINING

SOFTWARE QUALITY

THE . . . ILITIES



USABILITY (USER EXPERIENCE)



USABILITY (USER EXPERIENCE OR UX)

SIMPLICITY

FAMILIARITY

RECOGNITION

FLEXIBILITY

AFFORDANCE

SAFETY

RELIABILITY

It does what it is required to do without failing

Availability %	Downtime per year <small>[note 1]</small>	Downtime per month	Downtime per week	Downtime per day
55.55555555% ("nine fives")	162.33 days	13.53 days	74.92 hours	10.67 hours
90% ("one nine")	36.53 days	73.05 hours	16.80 hours	2.40 hours
95% ("one nine five")	18.26 days	36.53 hours	8.40 hours	1.20 hours
97%	10.96 days	21.92 hours	5.04 hours	43.20 minutes
98%	7.31 days	14.61 hours	3.36 hours	28.80 minutes
99% ("two nines")	3.65 days	7.31 hours	1.68 hours	14.40 minutes
99.5% ("two nines five")	1.83 days	3.65 hours	50.40 minutes	7.20 minutes
99.8%	17.53 hours	87.66 minutes	20.16 minutes	2.88 minutes
99.9% ("three nines")	8.77 hours	43.83 minutes	10.08 minutes	1.44 minutes
99.95% ("three nines five")	4.38 hours	21.92 minutes	5.04 minutes	43.20 seconds
99.99% ("four nines")	52.60 minutes	4.38 minutes	1.01 minutes	8.64 seconds
99.999% ("five nines")	5.26 minutes	26.30 seconds	6.05 seconds	864.00 milliseconds
99.9999% ("six nines")	3.16 seconds	262.98 milliseconds	60.48 milliseconds	8.64 milliseconds
99.99999% ("eight nines")	315.58 milliseconds	26.30 milliseconds	6.05 milliseconds	864.00 microseconds
99.999999% ("nine nines")	31.56 milliseconds	2.63 milliseconds	604.80 microseconds	86.40 microseconds

https://en.wikipedia.org/wiki/High_availability#Percentage_calculation

A "popular" metric of five nines reliability

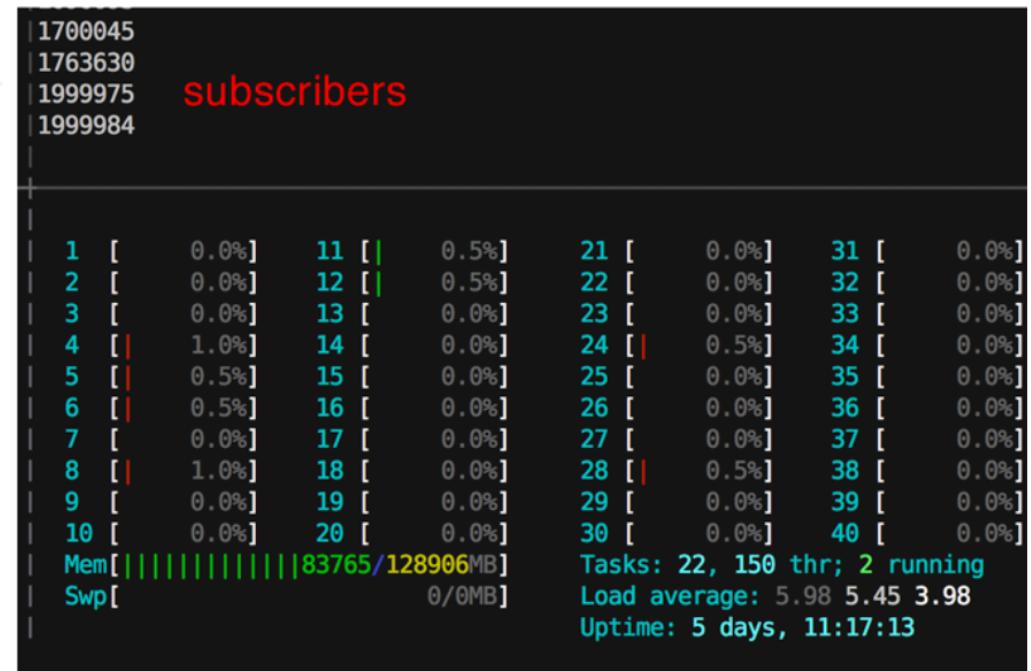
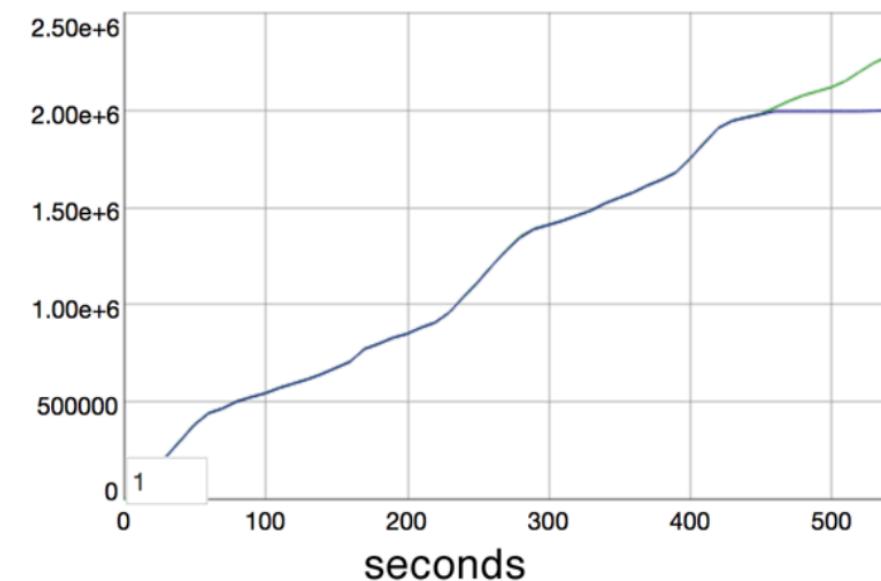
EFFICIENCY

← Reduces waste and maximized things like
CPU, I/O, Memory and Throughput

The Road to 2 Million Websocket Connections in Phoenix

By Gary Rennie · 2015-11-03 · v1.0.0

Simultaneous Users



<https://phoenixframework.org/blog/the-road-to-2-million-websocket-connections>

MAINTENANCE

*When faced with multiple options, choose the path that **makes future change easier**.*

<https://pragdave.me/blog/2014/03/04/time-to-kill-agile.html>

REUSABILITY



Its parts can be used in other projects, so reprogramming is not needed

How one programmer broke the internet by deleting a tiny piece of code

By Keith Collins • March 27, 2016



Re-use is not free

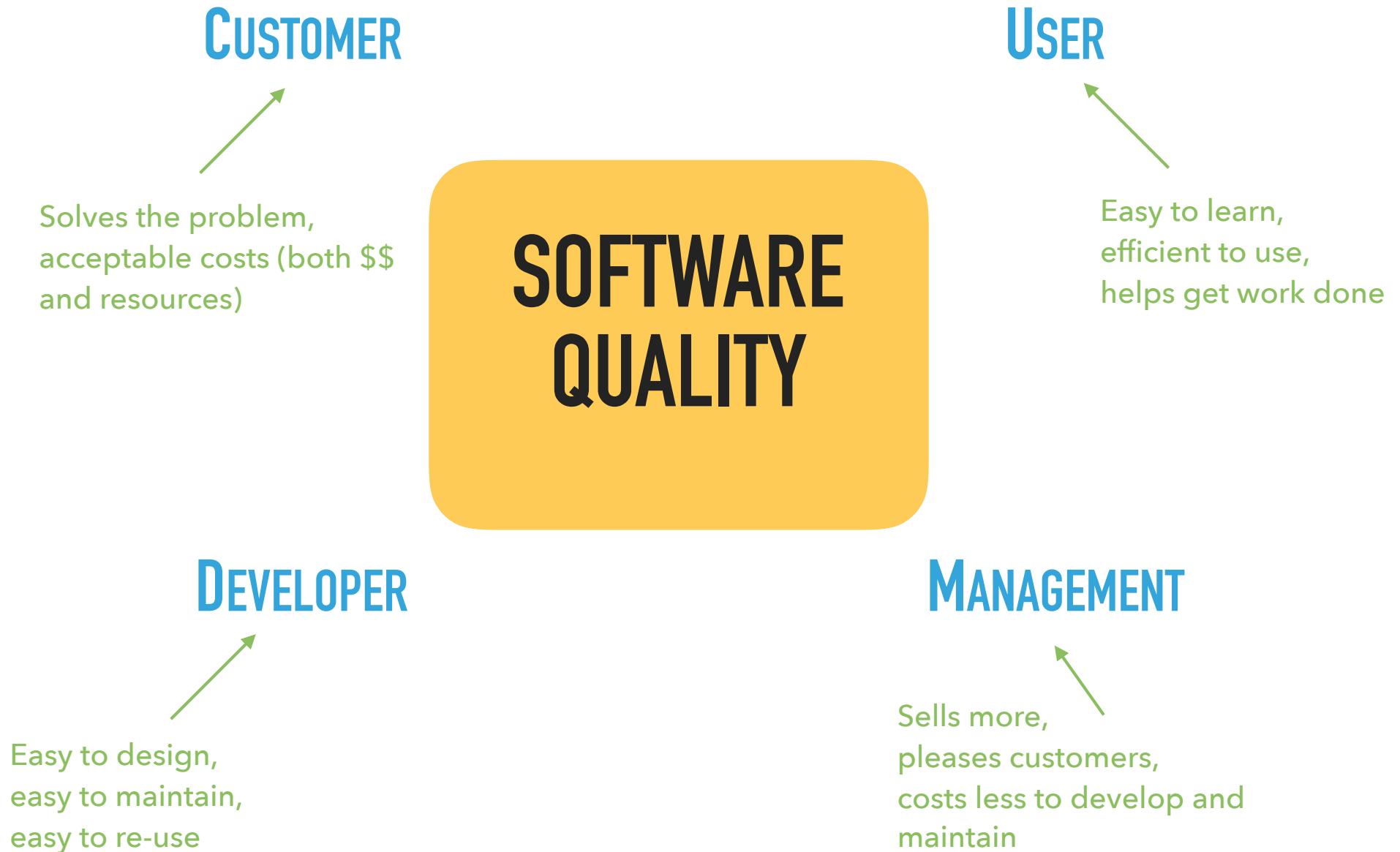
```
1 module.exports = leftpad;
2 function leftpad (str, len, ch) {
3     str = String(str);
4     var i = -1;
5     if (!ch && ch !== 0) ch = ' ';
6     len = len - str.length;
7     while (++i < len) {
8         str = ch + str;
9     }
10    return str;
11 }
12
```

<https://qz.com/646467/how-one-programmer-broke-the-internet-by-deleting-a-tiny-piece-of-code/>

```
08:08:00 pmueller ~/tmp/babel
$ node
> console.log(require('./packages/babel-core').guy),

```

<https://nodesource.com/blog/is-guy-fieri-in-your-node-js-packages/>



THE DIFFERENT QUALITIES CAN CONFLICT

- ▶ Increasing efficiency can reduce maintainability or reusability
- ▶ Increasing usability can reduce efficiency

Setting objectives for quality is a key engineering activity

- ▶ Design to meet the objectives
- ▶ Avoids 'over-engineering' which wastes money

MANY KINDS OF

**SOFTWARE ENGINEERING
PROJECTS**

Let's see if your React/Node/Docker/
Kubernetes code is still in production in 70
years



"Legacy code" often differs from its suggested alternative by **actually working and scaling.**

– Bjarne Stroustrup, creator of C++

MOSTLY EVOLUTIONARY OR MAINTENANCE PROJECTS

CORRECTIVE

ADAPTIVE

REENGINEERING

ENHANCEMENT

ENHANCEMENT PROJECTS

[Add a new idea](#)

[Recent](#)

[Trending](#)

[Popular](#)

FILTER BY STATUS

Already exists	11	2,549 VOTE
Will not implement	626	
Planned	51	
Shipped	754	

FILTER BY CATEGORY

Accounts	14	2,072 VOTE
Apps	12	
Autoscaling	0	
Backups	21	
Bandwidth	7	
Billing	123	
Block Storage	24	1,832 VOTE
CDN	5	
Cloud Firewalls	17	
Control Panel	10	
DigitalOcean General	1877	
DNS	46	1,803 VOTE
Domains	6	
Droplets	40	
Floating IPs	12	
GPUs	3	
Imaging	35	
	^	

Download snapshot and/or backup

It would be great if we could download a snapshot. I would use snapshot to replicate development environment on my local computer, I believe many others will do the same. Thanks

Created 12 months ago by Vito Meznaric
Imaging Planned 313

I know they cost a fortune, but I would love an Australian server!

Because most of the traffic going to my server would be Australia I would love it if you added one, I wouldn't mind paying extra for it either!

Created 12 months ago by Izaak
DigitalOcean General 405

Support Windows Server

No description provided.

Created 12 months ago by Anonymous
DigitalOcean General 460

Reserve IP Addresses / Make sure we can keep an IP if we recreate a droplet.

When destroying a droplet there's a 99% chance that you will get your IP back. If someone creates a droplet while destroying/creating a new one and steals your IP, it's a pain, even if the chances are <1%. This 1% is too much for some. An IP ...

Created 12 months ago by Dries Van Damme
Floating IPs 67

CORRECTIVE PROJECTS

- ▶ Fixing defects
 - ▶ No active work
 - ▶ “Maintenance mode”
 - ▶ Eventually *Adaptive* work required

CASE STUDY SITEVALET.COM

- ▶ Hotel CRM (Customer Relationship Manager)
- ▶ Active Development from 2006 - 2008
- ▶ Maintenance Mode afterwards
- ▶ Ruby on Rails 2.3,
- ▶ Ubuntu 8.04,
- ▶ Ruby 1.8.6

The screenshot shows the SiteValet Inn's basic information page. At the top, there's a green header bar with the SiteValet logo and the text "SiteValet Inn" and "View Public Website | Preview". Below this, on the left, is a sidebar with links for "Basic Info & Setup" (Contact, Location, Hours), "Home Page" (Your Business at a Glance), "About Page" (History, Architecture, Etc.), "Accommodations" (Rooms, In-Room Amenities), "Services & Amenities" (Property Amenities, Dining), "Rates & Booking" (Specials, Packages, Policies), and "Reputation". On the right, the main content area is titled "SiteValet Inn" and "Basic Information". It lists the following details:

Property Name	SiteValet Inn
Property Type	Inn
Your Unit Names	Rooms
Number of Floors	3
Rate Range	\$250 - \$500 / night
Time Zone	Atlantic Time (Canada)
Currency	Canadian Dollar

SITEVALET “MAINTENANCE”

- ▶ Manually clean logs every 3 months
- ▶ Manually reboot when image server got stuck
- ▶ *2 hours of maintenance PER YEAR*
- ▶ Lasted 8 years, until it was finally decommissioned

▼ General

Request URL: <http://my.sitevalet.com/inn/134/photo/create>

Request Method: POST

Status Code: 504 Gateway Time-out

Remote Address: 209.20.88.236:80

Referrer Policy: no-referrer-when-downgrade
<https://github.com/ambethia/smtp-tls>

TLS version deprecated by AWS S3 uploads, and underlying software was not maintained

– Response Headers view source

ambethia / smtp-tls

Code Issues 0 Pull requests 1

A ruby gem to enable TLS for Net::SMTP

on Apr 2, 2009

v1.1.2 ...

7d62b44 zip tar.gz

on May 21, 2008

v1.0 ...

0251c9e zip tar.gz

Hey! Don't blame the free software (thank you to Ambethia for the years of free upload software)

ADAPTIVE PROJECTS

- ▶ Changing the system in response to changes in
 - ▶ Operating system
 - ▶ Database
 - ▶ Rules and regulations
- ▶ Examples, adapting your project for
 - ▶ TLS 1.2 support
 - ▶ PHP 7 support
 - ▶ Ubuntu 18.04 support
 - ▶ Stripe V3 support

REENGINEERING OR PERFECTIVE PROJECTS

- ▶ Changing the system internally so it is more maintainable
- ▶ Examples,
 - ▶ Moving from AWS to Azure
 - ▶ Moving from Monolith to Microservices
 - ▶ Moving from Micro services back to Monolith

'GREEN FIELD' PROJECTS

NEW NEW

RARE

MINORITY

BECOMES
MAINTENANCE

SOFTWARE PROJECT

ACTIVITIES

REQUIREMENTS

- ▶ Domain analysis
- ▶ Defining the problem
- ▶ Requirements gathering
 - ▶ Obtaining input from as many sources as possible
- ▶ Requirements analysis
 - ▶ Organizing the information
- ▶ Requirements specification
 - ▶ Writing detailed instructions about how the software should behave

REQUIREMENTS AS A SALES PITCH

To-Do Groups

Jason Fried
Sep 12, 2017 - Notified 3 people

For 13 years we've done stuff like this...

iOS - To-dos - Morgan (Jul 17 - Aug 31)

- Morgan (Jul 17 - Aug 31) 1419
 - Jump Menus
 - Jump Menu
 - SMALL BATCH
 - Options menu redesign
 - OTHERS
 - Refactor Code
 - Forky sub-tasks
 - Use silent background push notifications to trigger a Hey! refresh
 - Present check-in questions in Hey! Reading List
 - Refactor Trix toolbar integration
 - Add to-do
 - Add support for horizontal dividers
 - Account creation detect and reward notifications
 - Design house account names
 - Or...

403 KB · View full-size · Download

What we're trying to do is create simple groups of to-dos within a single list. So we're hacking it. We're either creating "====Artificial Dividers=====" or prepending a batch to-dos with a label like "QA" or our old standby, the trusty "which is code for "maybe". Yeah it works, but it's messy, and it's not a refined pattern to share with customers. 13 years in, it's time to level up.

To-do groups

A couple weeks ago, Ryan and I spent an afternoon working out an idea out to make to-do groups first class citizens without adding noticeable complexity to the purity of today's to-do lists. We're pleased with where we landed, and we'd like to put it forward as a big batch project this cycle.

Here's how it works:

First, to-dos without groups will look identical to today's to-dos. We won't add any additional UI around them, no new buttons, etc...

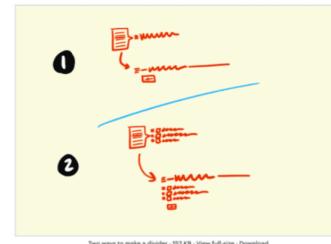


This means now you could make a list for a specific scope that had a section for design-specific to-dos, and QA-specific to-dos.



A group is defined by a divider. You can name a divider anything you want. It can't be checked off, it's not a to-do itself. Any to-dos you put below the divider become part of that group. And critically, groups always go below any ungrouped (loose) to-dos. This simplifies a bunch of behaviors, and prevents loose to-dos from being lost between groups. If it's loose, it's at the top.

You can create a divider in two ways...



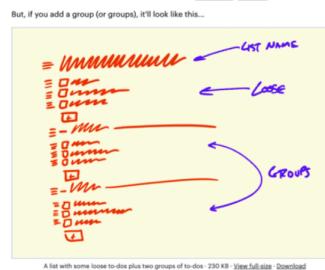
- Click the hamburger menu next to the list title. You'll get a menu. One of the items will be "Create a group" (or add a divider, exact language TBD). It'll then add a blank divider at the end of the loose to-dos, and scroll right to that point so you can name it.
- You can multi-select (shift key) multiple loose to-dos, and instantly create a group from those to-dos. If you do this, an unnamed divider will be created, and the to-dos will be placed under that divider.

Or...

BC3: Onboarding and Conversion (Morgan) - To-dos - Old Biz Model Phase-Out [1 week]

- Old Biz Model Phase-Out [1 week] 1010
 - QA: Michael B.
 - QA: On small screens current plan label alignment tweak
 - QA: On Android app plan selection page, jacked background color
 - QA: Could use a little spacing between confirmation button and fine print on upgrade confirmation page
 - QA: Discour combos read confusingly
 - QA: Frozen banner missing on /plans
 - QA: Seeing info for the \$49 plan, but unable to select it

QA - 463 KB · View full-size · Download



Key to this concept is that we'll add "Add a to-do" buttons to the bottom of each group (and the loose to-dos at the top). So once a divider exists, you can add to-dos directly to that section. If we didn't do this, and we only had one add-to-do button at the top or bottom, you'd have to add and then move into place. That's a hassle. It's much better to add it in place when we have places to add.

Completed to-dos will still be grouped into a single collection at the bottom of the entire list. If they were part of a group when you checked them off, we'll prepend the group name before the to-do like so...

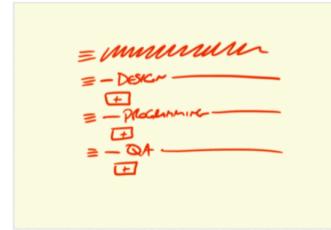


What we like about this concept overall is that it's very straightforward. It doesn't change existing to-do lists at all. No new permanent UI in the way, no behavior changes if you don't use groups.

But if you want to level up your organization - and we'd use the shit out of this - then you can add groups to a list. No groups within groups, no indenting - just as many one-level groups as you'd like.

Then there are other questions. Things like if you move the divider, do all the items move with it? I'd say yes since it's a group, but we'd have to work that out. Also, since you can't put groups above loose to-dos, we'd have to prevent dragging above a certain point, but that's double as well. I'm sure there are a few other questions as well, but we can work those out as we go.

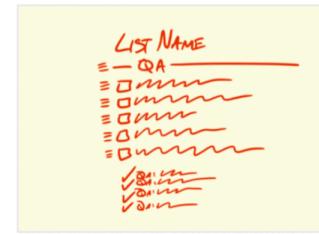
BONUS: Down the road we'd like to explore adding to-do templates to BC3. We have project templates, but now we're just talking to-do templates. You can imagine creating a template with no to-dos, but with groups in place. This is a process boost - people can organize projects in similar ways with just a little bit of structure like this...



Just add in place - 159 KB · View full-size · Download

If you uncheck a completed to-do from a group, it'll go back up to the group (assuming the group divider hasn't been deleted). 163 KB · View full-size · Download

Groups will also have their own perms. So you can click the title of a group/divider, and you'll see a new page with just the to-dos in that group...



Group perms for QA - 199 KB · View full-size · Download

https://basecamp.com/assets/books/shapeup/1.5/to-do_groups_pitch-ac9465339dfffb0b9cf634064b51b4d9336fe6d5c7410f7ed80abd1c1fbe5305.png

DESIGN

- ▶ Deciding how the requirements should be implemented, using the available technology
- ▶ Includes:
 - ▶ **Systems engineering:** Deciding what should be in hardware and what in software
 - ▶ **Software architecture:** Dividing the system into subsystems and deciding how the subsystems will interact
 - ▶ Detailed design of the internals of a subsystem
 - ▶ User interface design
 - ▶ Design of databases

MODELING

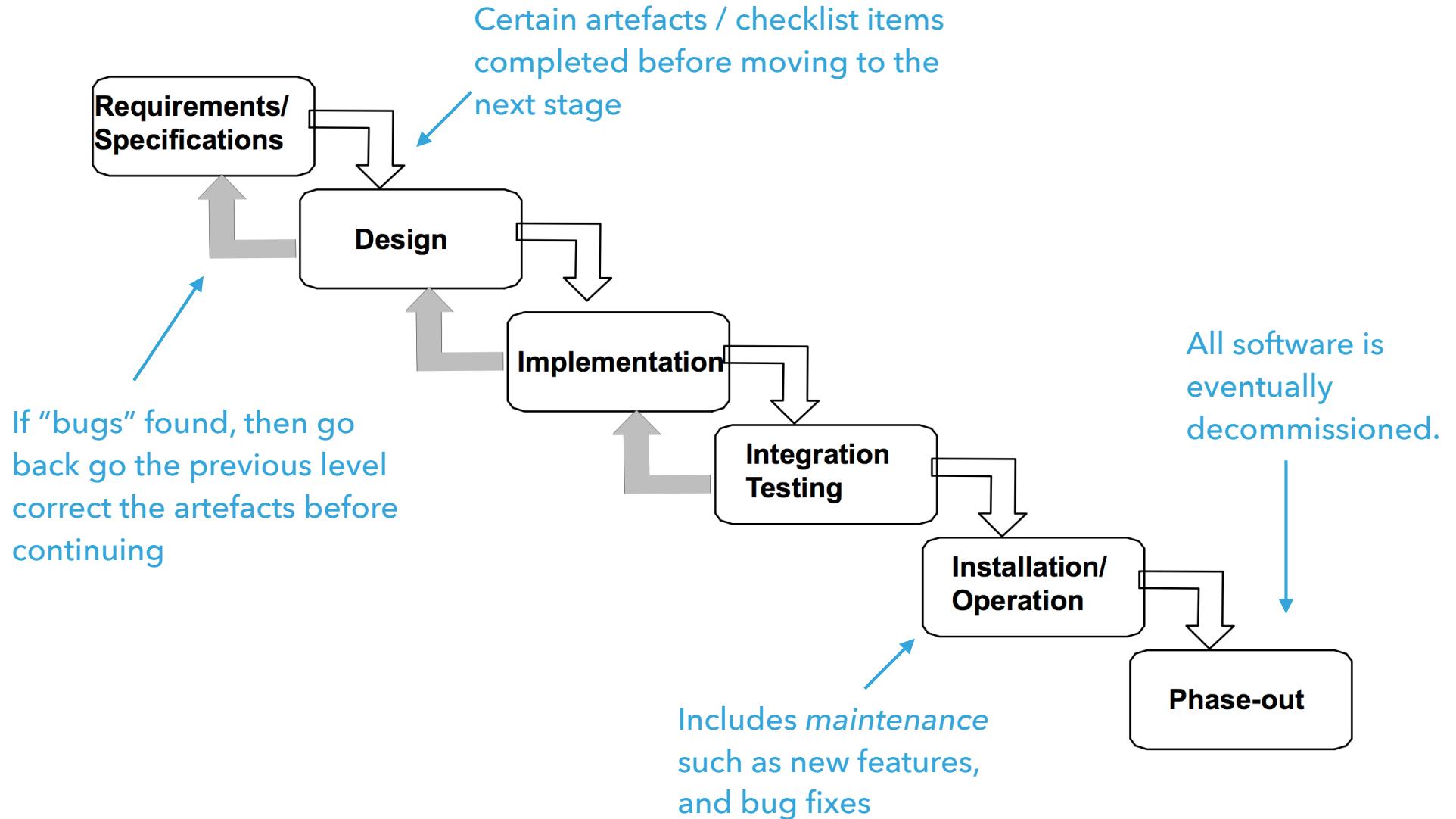
- ▶ Creating representations of the domain or the software
 - ▶ Use case modeling
 - ▶ Structural modeling
 - ▶ Dynamic and behavioural modeling

PROGRAMMING

- ▶ Test Driven Development
 - ▶ Test Last
 - ▶ Test Never
- ▶ Refactoring
- ▶ Integration
- ▶ Paired Programming
 - ▶ Mob Programming
- ▶ Code Review
- ▶ Pull Requests
- ▶ Trunk Base versus Feature Based Dev

MANAGING THE PROCESS

- ▶ Project Management
- ▶ Product Management
- ▶ Operations (“DevOps”)



THE PROCESS THAT WASN'T MEANT TO BE

"In my experience, the simpler method has never worked on large software development efforts..."

- Royce, Winston (1970), "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON 26 (August): 1-9

Yet many have cited this work as the basis for *do waterfall!*

Referring to Waterfall method

Goes on to describe a *better* alternative that really isn't waterfall at all.

ROYCE'S FIVE STEPS

1. Program Design Comes First

Avoids ivory tower architectures and analysis that simply cannot be efficiently implemented

2. Document the Design

Before code there is still design, and important things are written down

3. Do it Twice

"Plan to throw first version away" (Brook's Mythical man month)

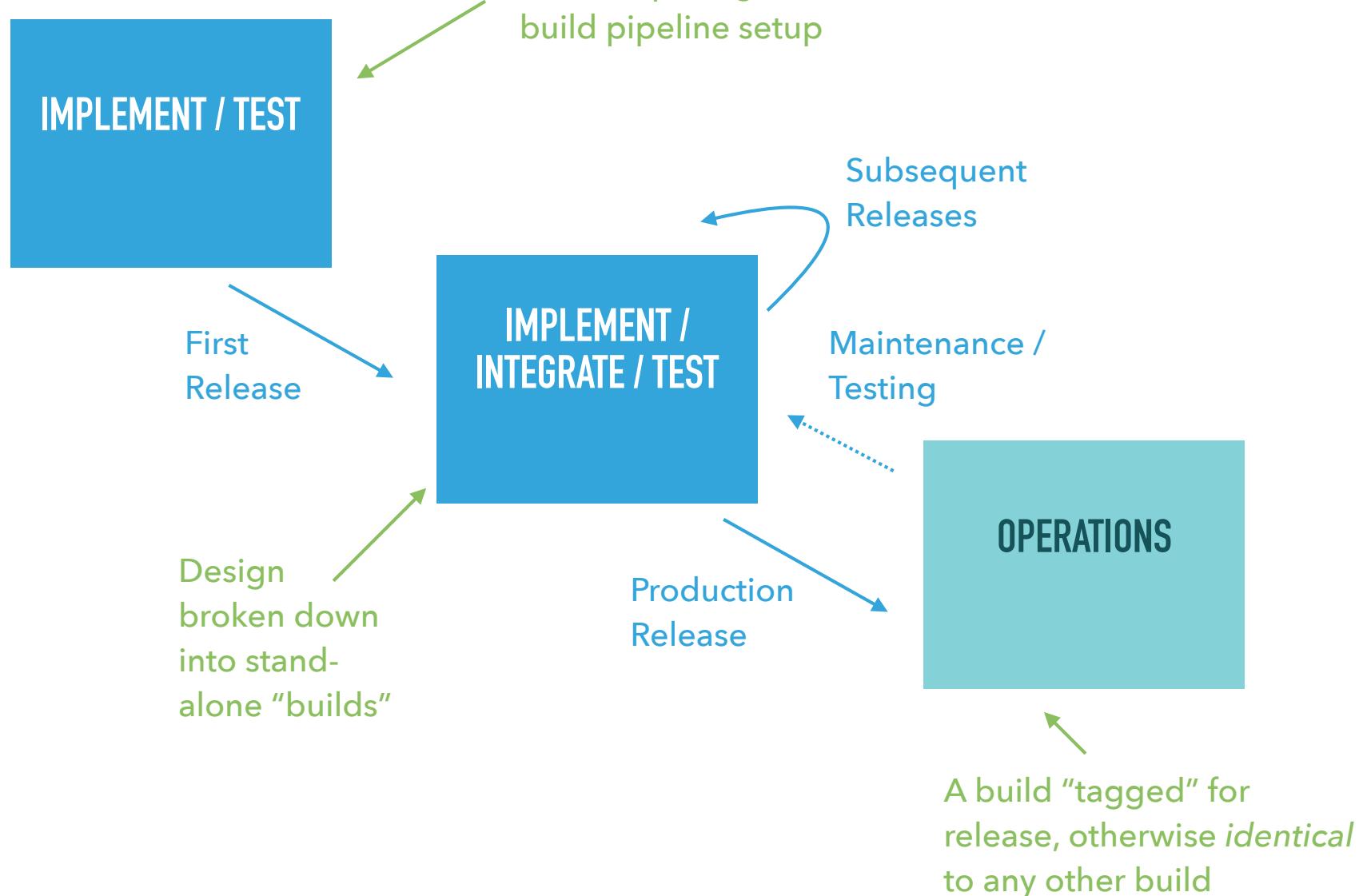
4. Plan, Control and Monitor Testing

Early and incremental testing. Software is ultimately about automation, so we should *dog food* that with testing.

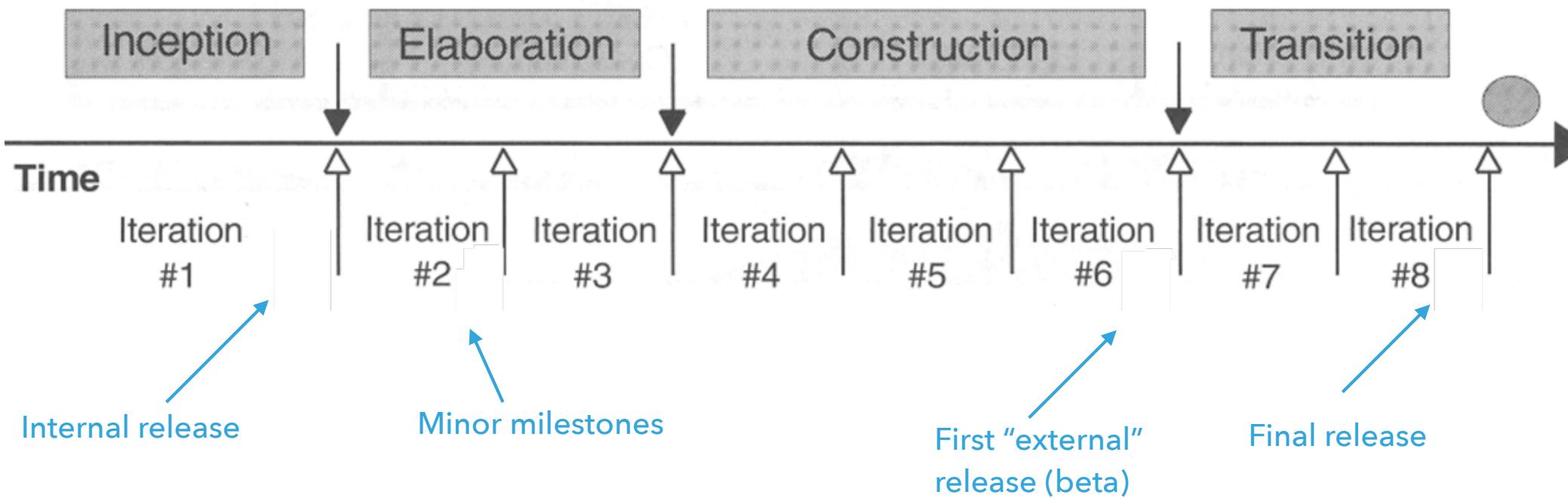
5. Involve the Customer

Something lacking in typical Waterfall

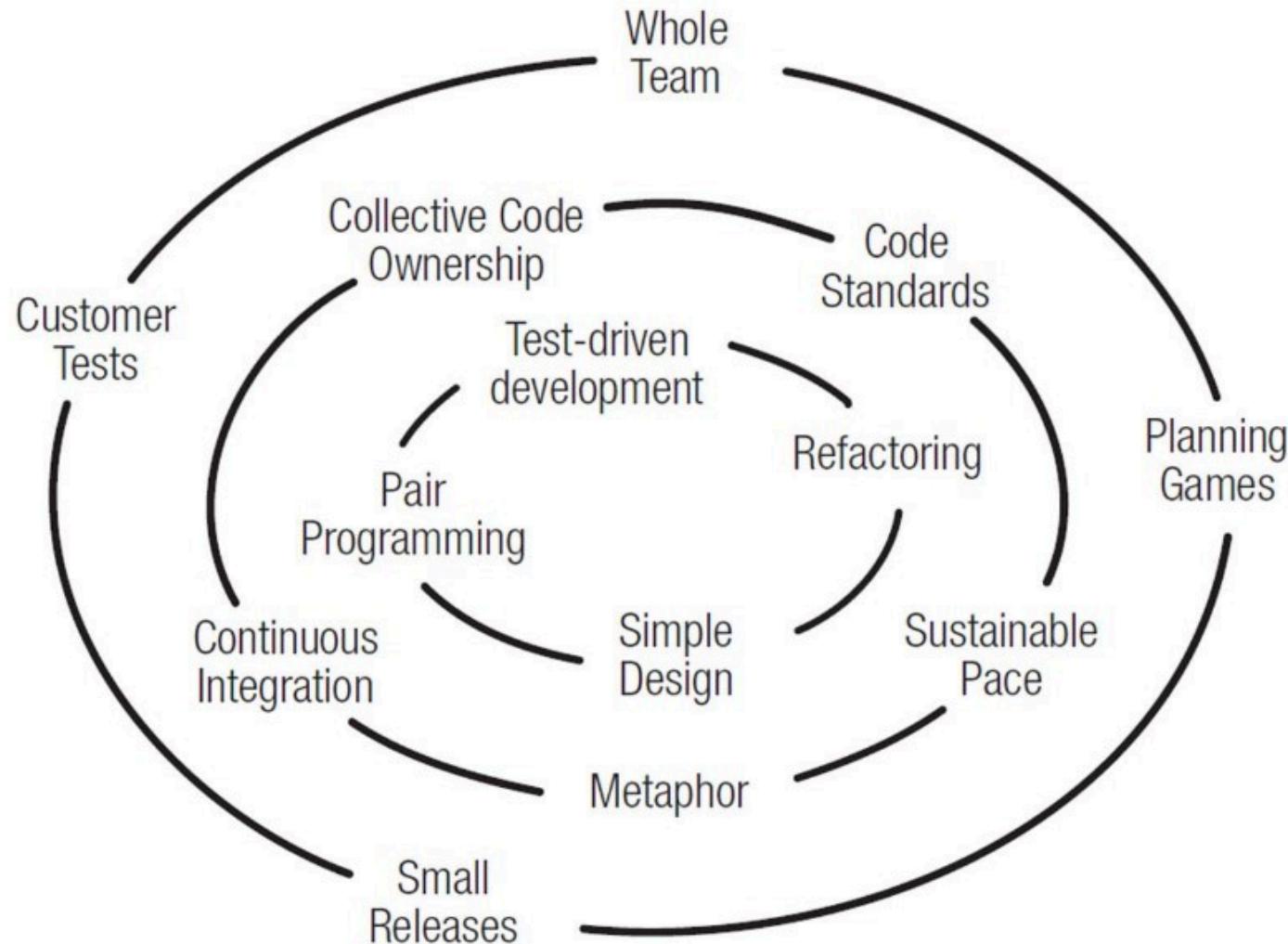
ITERATIVE AND INCREMENTAL



TIMELINES



EXTREME PROGRAMMING



SCRUM

Inputs from Executives,
Team, Stakeholders,
Customers, Users



Product Owner

The Team



Product
Backlog

Team selects starting at top as much as it can commit to deliver by end of Sprint

Sprint
Planning
Meeting



Sprint
Backlog

Scrum
Master



Burndown/up
Charts

Every
24 Hours

1-4 Week
Sprint



Daily Scrum
Meeting



Sprint Review



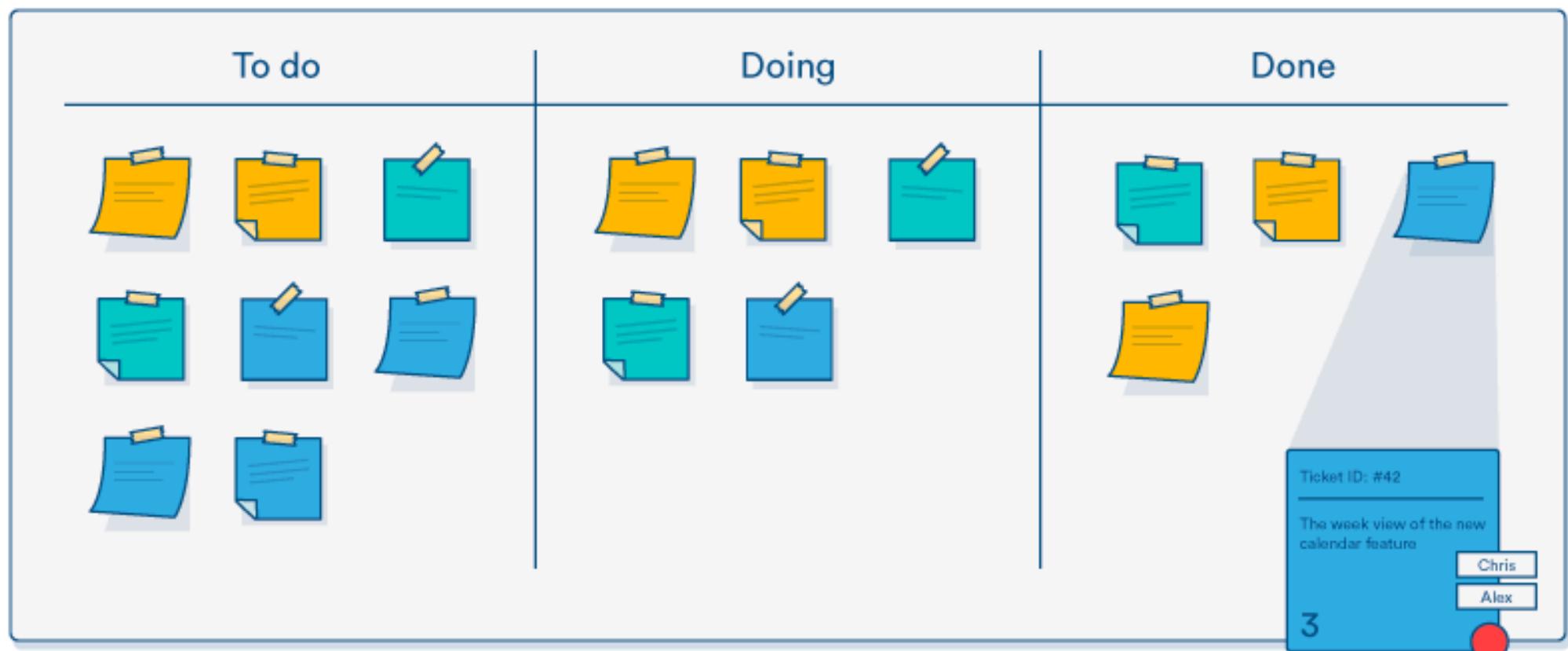
Finished Work

Sprint end date and team deliverable do not change



Sprint
Retrospective

KANBAN



TRYING SOMETHING NEW... friday830

[piazza.com/uottawa.ca/fall2019/
seg2105a/home](https://piazza.com/uottawa.ca/fall2019/seg2105a/home)

REFERENCES

- ▶ <https://www.slideshare.net/YuriyKravchenko/software-development-life-cycle-sdlc-15545167>
- ▶ https://twitter.com/abt_programming/status/569243921718611969
- ▶ <https://stackoverflow.com/questions/947874/what-is-functional-decomposition>
- ▶ <https://info.obsglobal.com/blog/2013/01/software-history-waterfall-the-process-that-wasnt-meant-to-be>
- ▶ <https://www.youtube.com/watch?v=vqz8ND-N1hc>
- ▶ https://www.youtube.com/watch?v=dOg41YdZ_as