

# Build Software Safely!

17-313: Foundations of Software Engineering

<https://cmu-313.github.io>

Michael Hilton and Josh Sunshine

Spring 2026

# Learning Goals

- Learn to discuss risk in a project
- Strategize about ways to mitigate risk
- Learn to get early feedback to reduce risk
- Find ways to catch our technical errors

# Administrivia.

- P2B Due Tuesday, February 10, 2026
- Midterm review session TBD
- Midterm Thu Feb 26

# Smoking Section

- Last full row



# Risk

# Risk

Tony Webster   
@webster

Follow

I appreciate the honesty.

**Pick a password**

Don't reuse your bank password, we didn't spend a lot on security for this app.  
At least 6 characters

**Continue**

8:20 PM - 15 Sep 2018

5,868 Retweets 15,672 Likes

58 5.9K 16K

# Definition: Risk

Risk is a measure of the potential inability to achieve overall program objectives within defined cost, schedule, and technical constraints.



7

# Risk is defined by two key components



The probability (or likelihood) of failing to achieve a particular outcome



The consequences (or impact) of failing to achieve that outcomes

# Internal vs. External Risk



Risks that we **can control**



Risks that we **cannot** control

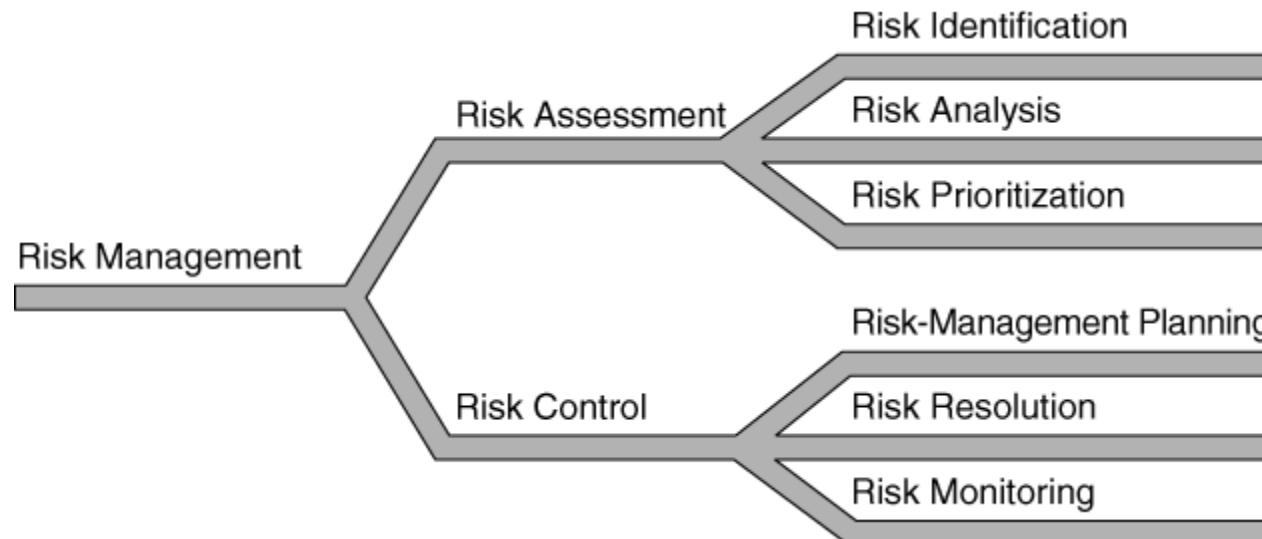
# Levels of Risk Management

1. Crisis management: Fire fighting; address risks only after they have become problems.
2. Fix on failure: Detect and react to risks quickly, but only after they have occurred.
3. Risk mitigation: Plan ahead of time to provide resources to cover risks if they occur, but do nothing to eliminate them in the first place.
4. Prevention: Implement and execute a plan as part of the software project to identify risks and prevent them from becoming problems.  
10
5. Elimination of root causes: Identify and eliminate factors that make it possible for risks to exist at all.

# Levels of Risk Management

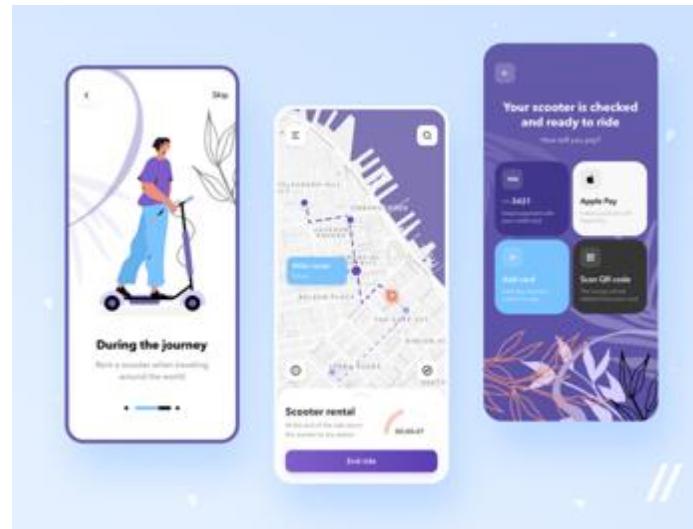
1. Crisis management: Fire fighting; address risks only after they have become problems.
2. Fix on failure: Detect and react to risks quickly, but only after they have occurred.
3. Risk mitigation: Plan ahead of time to provide resources to cover risks if they occur, but do nothing to eliminate them in the first place.
4. Prevention: Implement and execute a plan as part of the software project to identify risks and prevent them from becoming problems.<sup>11</sup>
5. Elimination of root causes: Identify and eliminate factors that make it possible for risks to exist at all.

# Risk Management



# Team Exercise: Risk Identification

- What risks exist for the scooter app?



13

# Risk assessment matrix



TABLE III. Risk assessment matrix

		RISK ASSESSMENT MATRIX			
		Catastrophic (1)	Critical (2)	Marginal (3)	Negligible (4)
SEVERITY PROBABILITY \		Frequent (A)	Probable (B)	Occasional (C)	Remote (D)
High	High	High	Serious	Medium	Medium
High	High	High	Serious	Medium	Medium
High	Serious	Serious	Medium	Low	Low
Serious	Medium	Medium	Medium	Low	Low
Medium	Medium	Medium	Medium	Low	Low
Eliminated (F)		Eliminated			

- MIL-STD-882E

<https://www.system-safety.org/Documents/MIL-STD-882E.pdf>

# Aviation failure impact categories

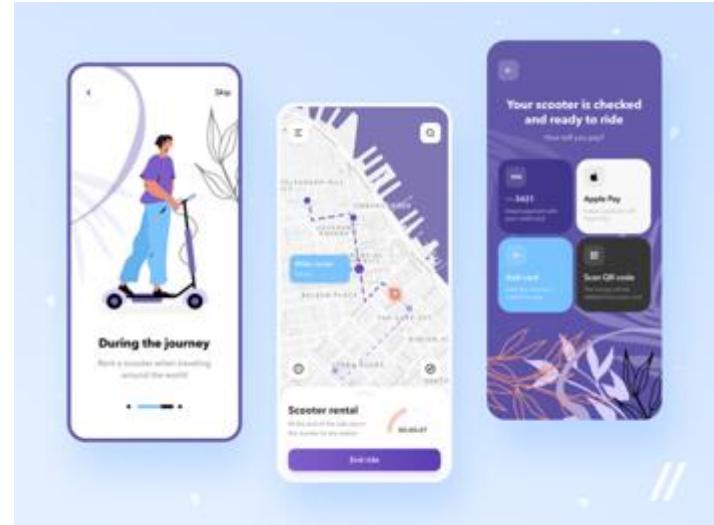
- No effect – failure has no impact on safety, aircraft operation, or crew workload
- Minor – failure is noticeable, causing passenger inconvenience or flight plan change
- Major – failure is significant, causing passenger discomfort and slight workload increase
- Hazardous – high workload, serious or fatal injuries
- Catastrophic – loss of critical function to safely fly and land

# Risk Analysis

Risk	Probability (%)	Size of Loss (weeks)	Risk Exposure (weeks)
Overly optimistic schedule	50%	5	2.5
Additional features added by marketing (specific features unknown)	35%	8	2.8
Project approval takes longer than expected	25%	4	1.0
Management-level progress reporting takes more developer time than expected	10%	1	0.1
New programming tools do not produce the promised savings	30%	5	1.5
...	...	...	...
<b>Total</b>			12

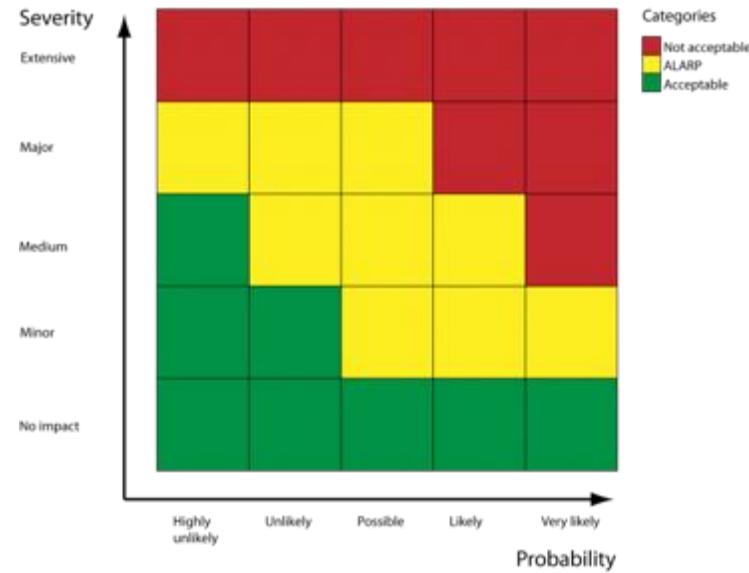
# Exercise: Risk Analysis

- What is the risk severity for your scooter app?



# Risk Prioritization

## Focus on risks with the highest exposure



18

# Risk Control

- What steps can be taken to avoid or mitigate the risk?
- Can you better understand and forecast the risk?
- Who will be responsible for monitoring and addressing the risk?
- Have risks evolved over time?
- Bake risks into your schedule
  - Don't assume that nothing will go wrong between now and the end of the semester!

# DECIDE Model



**D**etect that the action necessary

**E**stimate the significance of the action

**C**hoose a desirable outcome

**I**dentify actions needed in order to achieve the chosen option

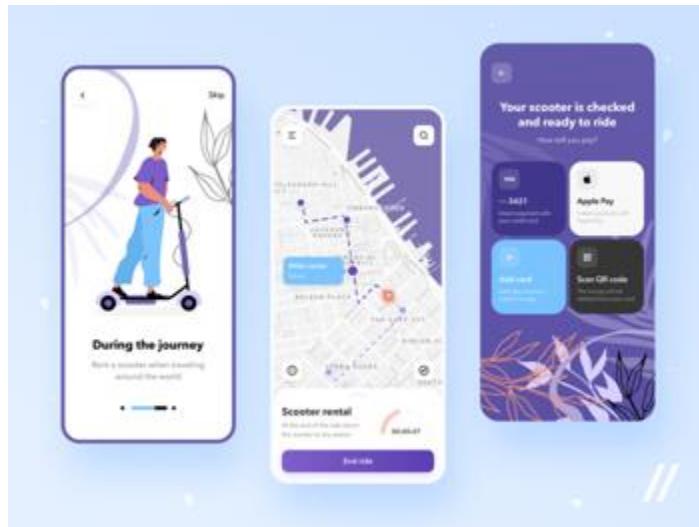
**D**o the necessary action to achieve change

**E**valuate the effects of the action

[https://www.faa.gov/regulations\\_policies/handbooks\\_manuals/aviation/media/FAA-H-8083-2.pdf](https://www.faa.gov/regulations_policies/handbooks_manuals/aviation/media/FAA-H-8083-2.pdf)

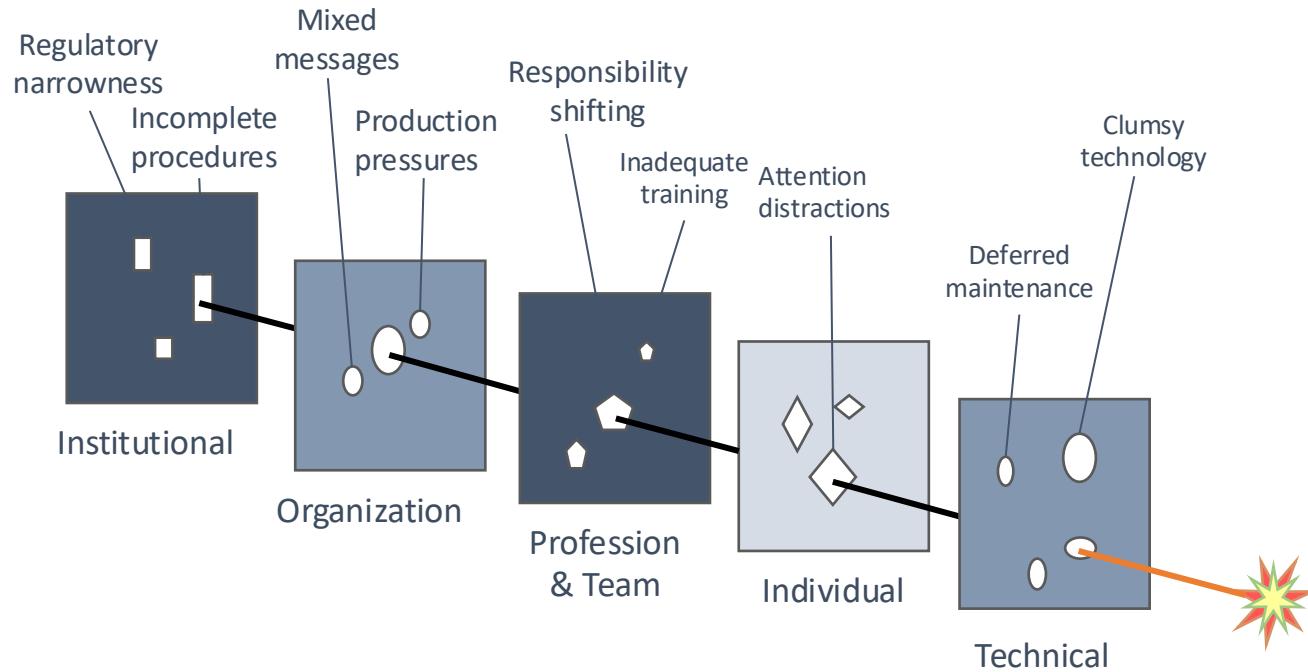
# Discussion: Risk Elimination and Mitigation

- How can you eliminate/mitigate risk for your scooter app?



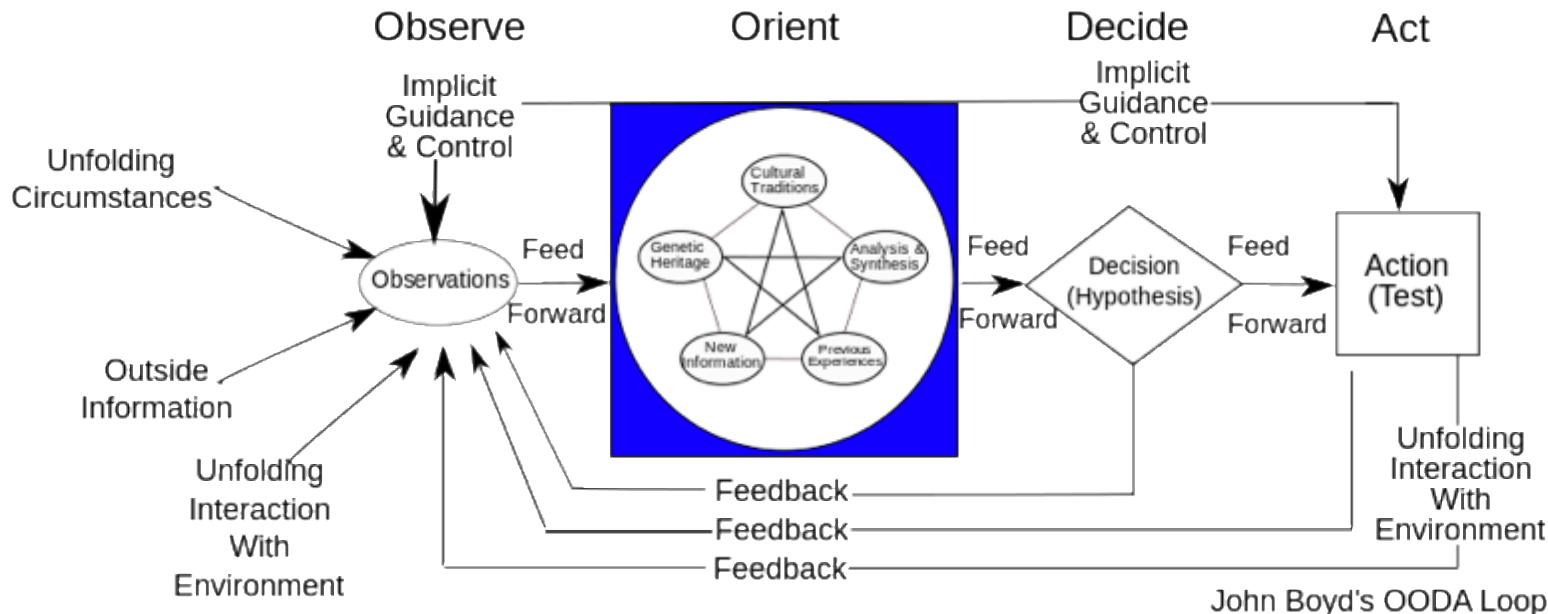
21

# The Swiss cheese model



Modified from Reason, 1999, by R.J. Crook

# OODA Loop



By Patrick Edwin Moran - Own work, CC BY 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=3904554>

# No matter what you do

- Some idiots won't follow your rules 😊



# Pre-mortems

- "unlike a typical critiquing session, in which project team members are asked what might go wrong, the premortem operates on the assumption that the 'patient' has died, and so asks what did go wrong."

Project Management

## Performing a Project Premortem

by Gary Klein

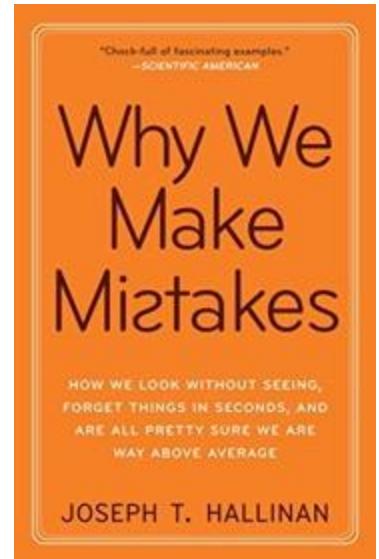
From the Magazine (September 2007)



**Summary.** Reprint: FOTO8A In a premortem, team members assume that the project they are planning has just failed—as so many do—and then generate plausible reasons for its demise. Those with reservations may speak freely at the outset, so that the project can be... [more](#)

# Why do we make mistakes?

26



# Generalization

- ...in the words of psychologist Tom Stafford, we can't find our typos because we're engaging in a high-level task in writing. Our brains generalize simple, component parts to focus on complex tasks, so essentially we can't catch the small details because we're focused on a large task.

27

<https://medium.com/swlh/why-we-miss-our-own-typos-96ab2f06afb7>

Boredom can give rise to errors, adverse patient events, and decreased productivity—costly and unnecessary outcomes for consumers, employees, and organizations alike. As a function of boredom, individuals may feel over-worked or under-employed, and become distracted, stressed, or disillusioned. Staff who are bored also are less likely to engage with or focus on their work.

Original Articles

## Boredom in the Workplace: Reasons, Impact, and Solutions

Michelle Cleary , PhD, RN, Jan Sayers, PhD, RN, Violeta Lopez, PhD, RN & Catherine Hungerford, PhD, RN  
Pages 83-89 | Received 24 Jun 2015, Accepted 13 Aug 2015, Published online: 10 Feb 2016

 Download citation  <https://doi.org/10.3109/01612840.2015.1084554> 

 Full Article  Figures & data  References  Citations  Metrics  Reprints & Permissions  Get access

### Abstract

Boredom in the workplace is not uncommon, and has been discussed widely in the academic literature in relation to the associated costs to individuals and organizations. Boredom can give rise to errors, adverse patient events, and decreased productivity—costly and unnecessary outcomes for consumers, employees, and organizations alike. As a function of boredom, individuals may

Related rese

People also read 

Boredom at work spillover model c work motivation boredom > 

# Cognitive Load

- ..." students who switch back and forth between attending to a classroom lecture and checking e-mail, Facebook, and IMing with friends"



Computers & Education 62 (2013) 24–31

Contents lists available at SciVerse ScienceDirect

Computers & Education

journal homepage: [www.elsevier.com/locate/compedu](http://www.elsevier.com/locate/compedu)

Laptop multitasking hinders classroom learning for both users and nearby peers

Faria Sana<sup>a</sup>, Tina Weston<sup>b,c</sup>, Nicholas J. Cepeda<sup>b,c,\*</sup>

<sup>a</sup> McMaster University, Department of Psychology, Neuroscience, & Behaviour, 1280 Main Street West, Hamilton, ON L8S 4K1, Canada

<sup>b</sup> York University, Department of Psychology, 4700 Keele Street, Toronto, ON M3J 1P3, Canada

<sup>c</sup> York University, LaMarsh Centre for Child and Youth Research, 4700 Keele Street, Toronto, ON M3J 1P3, Canada

---

ARTICLE INFO

Article history

---

ABSTRACT

Laptop use compromised the academic outcomes of both of computer multitaskers themselves as well as

# Can we remove human error?

30

# catch Can we ~~remove~~ human error?

Can we catch human error before we ship our code?<sup>31</sup>

Can we automate tasks to prevent problems?



Distinctio sexta Tractatus secundus.

**D**ba notare circa a la figura q' faccio pofta cb fini fodiiti numeri ò la linea prima quale comenza da, e una fin a. 10. ducendo cofi: 1.2.3.4.5.6.7.8.9.10. El numero da, secondo spacio al numero del primo referira bauerai la prima specie de la propozion multipli cioe dupla. E fe al primo referir adi numero del ro so spacio tra- ras la seconda specie de la multiplici cioe triplice. E cofi sequiendo in tante le altre righe de los rati simili trovansi. Q'ba fe al numero del secondo spacio el numero del terzo spacio cioe 1.2.3. Abstratta prima specie te la propozione superparticulare cioe ferma altera. E fe al terzo el quarto cito: 4.5.6. sequenteria. E fe al quarto lo quinto cito: 5.6.7.8. capa quaria e cofi in latere fogni. Q'ba fe al numero del terzo fe adio compreherai el numero del quarto spacio cito: 5.6.7.8. bauerai la prima specie de la propozione superparticulare cito super partita tertia. E fe al numero del quarto spacio el numero del septimo referirai cito: 7.8.9. bauerai la seconda specie de la propozione superparticulare cioe supertripartita qua- tura. E fe al numero del quinto el sesto del nono cito nonne a.5. farai la terza specie de la sup partita cioe super quadripartita quinaria. E fe al numero del secondo el numero del q'lo cito: 5.6.7. farai la prima specie de la propozione multipli superparticulare cito de feliqualetra. E fe a quel medesimo el septimo cito: 7.8.9. tripla fequaletra. Q'ba fe al nu- mero del terzo si compara el numero de locatio cito: 8.9.10. la prima specie de la propozione multipli superparticulare cioe dupla supertripartita tertia. E collipotrai per te puo- cedere fe la tabula sira magioce. E fe al quarto undecimo quando in fosse faresti latita specie detta dupla supertri partita quartas cito: 11. quando più oltra volesti proceder e cofi commo abbiamo detto de li numeri posti nella prima riga secondo li medesimi gbi comparando ancora le linee inferiori quelle medesime specie te daranno che fin la prima barbaute, spero ni per te sequirai. 7.

**D**etermino etiam q' uia in denominare molte specie de proportioni di qui non importa altro. A te pratico s'ieno a più comodamente proferire q'ci trovano. Et est: vt supza de multiplicando integros numeros decimam syllabica addicere. Si como dicimus del via e del sia che fuiano al- core re. Q'ba el sub cb a causare le specie de la meiora inequality li poneone a quelle giore inequality. E il meiora popolico e cofi li super in più specie interpolato. Io no mis- De proportionalitatibus tr. 2. acte d'istud. arti. p.

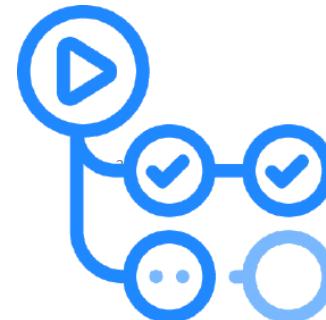
Intendo a bastanza de le proportioni parlato e quelle quale fine a le

1	2	3	4	5	6	7	8	9	10
4	9	27	81	243	729	2187	6561	19683	59049
8	19	51	153	459	1377	4131	12393	37179	111537
16	38	114	342	1026	3078	9234	27692	83076	249228
32	76	228	704	2112	6336	19008	57024	171072	513216
64	152	456	1408	4224	12672	38016	114048	342144	1026432
128	304	912	2816	8448	25344	76088	228240	704720	2101440
256	608	1824	5632	16896	51152	156480	469408	142320	424960
512	1216	3648	11296	34256	102768	308960	926880	285064	855192
1024	2432	7296	22992	70512	211536	634512	193360	580192	1760576
2048	4864	14592	45984	14192	42576	125920	383840	115152	350456
4096	9728	29184	91920	28384	85152	255840	767520	231456	704368
8192	19456	58368	183840	55152	165152	495440	149280	46384	141152
16384	38912	116736	367680	115152	345152	1035840	318560	98384	302304
32768	77824	233472	735360	22756	69152	207160	63616	20116	6036
65536	155648	466944	1470720	45112	13824	41448	13128	40416	12128
131072	311296	933888	2911440	90224	27648	85152	26384	80576	24152
262144	622592	1867776	5823360	180448	55296	17056	51456	15432	4608
524288	1245184	3735552	11466720	360896	110592	34112	104384	31368	9216
1048576	2490368	7471104	23413440	721792	221184	6832	20876	6296	1852
2097152	4980736	14942208	46826880	1443584	442368	13664	41672	12584	3712
4194304	9961472	29884416	93653760	2887168	884736	27328	83344	25056	7424
8388608	19922944	59768832	187307520	5774336	1769472	52160	160032	49360	14816
16777216	39845888	119537664	374615040	11548672	3538944	106480	320064	98160	30432
33554432	79691776	238275328	748830080	23097344	7078888	212960	640128	196320	60864
67108864	159383520	476550656	1497660160	46194688	14157776	425920	1280256	392640	121760
134217728	318767040	953101296	2995320320	92389376	28315552	85152	2560512	80576	24352
268435456	637534080	1906202592	5985640640	184778752	56631104	17056	5121024	161152	4852
536870912	1275068160	3812405184	11971281280	369537504	11329232	34112	104384	31368	9216
1073741824	2550136320	7624810368	23943563200	738575008	22957464	6832	20876	6296	1852
2147483648	5100272640	1524962072	47887126400	1477150016	45874928	13664	41672	12584	3712
4294967296	1020054520	3049924144	95774252800	2954300032	89549856	27328	83344	25056	7424
8589934592	2040109040	6099848288	191548505600	5854600016	17859968	52160	160032	49360	14816
17179869184	4080218080	12199696576	383096011200	1154867200	35389440	106480	320064	98160	30432
34359738368	8160436160	24399393152	765184033600	2309734400	70788880	212960	640128	196320	60864
68719476736	16320872320	48798786304	1520368067200	4619468800	141577760	425920	1280256	392640	121760
137438953472	32641744640	97597572608	3040736134400	9238937600	283155520	85152	2560512	80576	24352
274877906944	65283489280	195195145216	6081472338400	18477875200	566311040	17056	5121024	161152	4852
549755813888	130566978560	390387890432	1216344667200	36953750400	113292320	34112	104384	31368	9216
1099511627776	261133957120	780775780864	2432688664000	73857500800	229574640	6832	20876	6296	1852
2199023255552	522267914240	156155156168	4865377328000	147715001600	458749280	13664	41672	12584	3712
4398046511104	1044535828480	312310312336	9730754656000	295430003200	895498560	27328	83344	25056	7424
8796093022208	2089071656960	624620624672	19461509312000	585460001600	178599680	52160	160032	49360	14816
17592186044416	4178143313920	12492012464	38923018624000	115486720000	353894400	106480	320064	98160	30432
35184372088832	8356286627840	24984024928	77846037248000	230973440000	707888800	212960	640128	196320	60864
70368744177664	16712573255680	49968049456	15569207448000	461946880000	1415777600	425920	1280256	392640	121760
140737488355328	33425146511360	99936098912	31138414896000	923893760000	2831555200	85152	2560512	80576	24352
281474976710656	66850293022720	199872197824	62276829792000	1847787520000	5663110400	17056	5121024	161152	4852
562949953421312	133700586045440	399744395648	124553659584000	3695375040000	1132923200	34112	104384	31368	9216
1125899906842624	267401172090880	799488791296	249107319168000	7385750080000	2295746400	6832	20876	6296	1852
2251799813685248	534802344181760	1598977582592	498214638336000	14771500160000	4587492800	13664	41672	12584	3712
4503599627370496	1069604688363520	3197955165184	996429276672000	29543000320000	8954985600	27328	83344	25056	7424
9007199254740992	2139209376727040	6395910330368	1992858553344000	58546000160000	1785996800	52160	160032	49360	14816
18014398509481984	4278418753454080	12791820660736	3985717106688000	11548672000000	3538944000	106480	320064	98160	30432
36028797018963968	8556837506908160	25583641321472	7971434213376000	23097344000000	7078888000	212960	640128	196320	60864
72057594037927936	17113675013816320	51167282642944	15942868426752000	46194688000000	14157776000	425920	1280256	392640	121760
14411518807985968	34227350027632640	102334565285888	31885736853504000	92389376000000	28315552000	85152	2560512	80576	24352
28823037615971936	68454700055265280	20466913057176	63771473707008000	18477875200000	56631104000	17056	5121024	161152	4852
57646075231943872	136909400110530560	40933826114352	127542947414016000	36953750400000	11329232000	34112	104384	31368	9216
115292150463887744	273818800221061120	81867652228704	255085894828032000	73857500800000	22957464000	6832	20876	6296	1852
230584300927775488	547637600442122240	16373530445744	510171789656064000	147715001600000	45874928000	13664	41672	12584	3712
46116860185555096	1095275200884244480	32747060891488	1020343579312128000	295430003200000	70788880000	212960	640128	196320	60864
92233720371110192	2190550401768488960	65494121782976	2040687158624256000	585460001600000	141577760000	425920	1280256	392640	121760
184467440742220384	4381100803536977920	130988243565952	4081374317248512000	115486720000000	283155520000	85152	2560512	80576	24352
368934881484440768	8762201607073955840	26197648713188	8162748634497024000	230973440000000	566311040000	17056	5121024	161152	4852
737869762968881536	17524403214147911680	52395297426376	16325497268994048000	461946880000000	1132923200000	34112	104384	31368	9216
1475739525937763072	35048806428295823360	10479059485272	32650994537988096000	923893760000000	2295746400000	6832	20876	6296	1852
2951479051875526144	70097612856591646720	20958118970544	65301989075976192000	1847787520000000	4587492800000	13664	41672	12584	3712
5902958103751052288	14019522571318329360	41916237941088	130603978151952384000	3695375040000000	9074985600000	212960	640128	196320	60864
11805916207502104576	28039045142636658720	83832475882176	261207956303880768000	7385750080000000	1415777600000000	425920	1280256	392640	121760
23611832415004209152	56078090285273317440	167664951764352	522415912607761536000	14771500160000000	2831555200000000	85152	2560512	80576	24352
47223664830008418304	112156180570546634880	335329903528704	1044838245215523072000	29543000320000000	5663110400000000	17056	5121024	161152	4852
94447329660016836608	224312361141093269760	670659807057408	2089676490431046144000	58546000160000000	11329232000000000	34112	104384	31368	9216
188894659320033673216	448624722282186539520	1341319614114816	4179352980862092288000	11548672000000000	22957464000000000	6832	20876	6296	1852
377789318640067346432	897249444564373079040	2682639228229632	8358705961724184576000	23097344000000000	45874928000000000	13664	41672	12584	3712
755578637280014673264	1794498889128746158080	5365278456459264	16717411923483769152000	46194688000000000	90749856000000000	212960	640128	196320	60864
151115727456002934652	3588997778257492316160	1073055691291852	33434823846967538304000	92389376000000000	14157776000000000	425920	1280256	392640	121760
302231454912005869304	7177995556514984632320	2146111382583704	66869647693935076608000	18477875200000000	36953750400000000	85152	2560512	80576	24352
604462909824001738608	14355991113029769264640	4292222765167408	133739295387870153216000	36953750400000000	73857500800000000	17056	5121024	161152	4852
1208925819648034677216	287119								

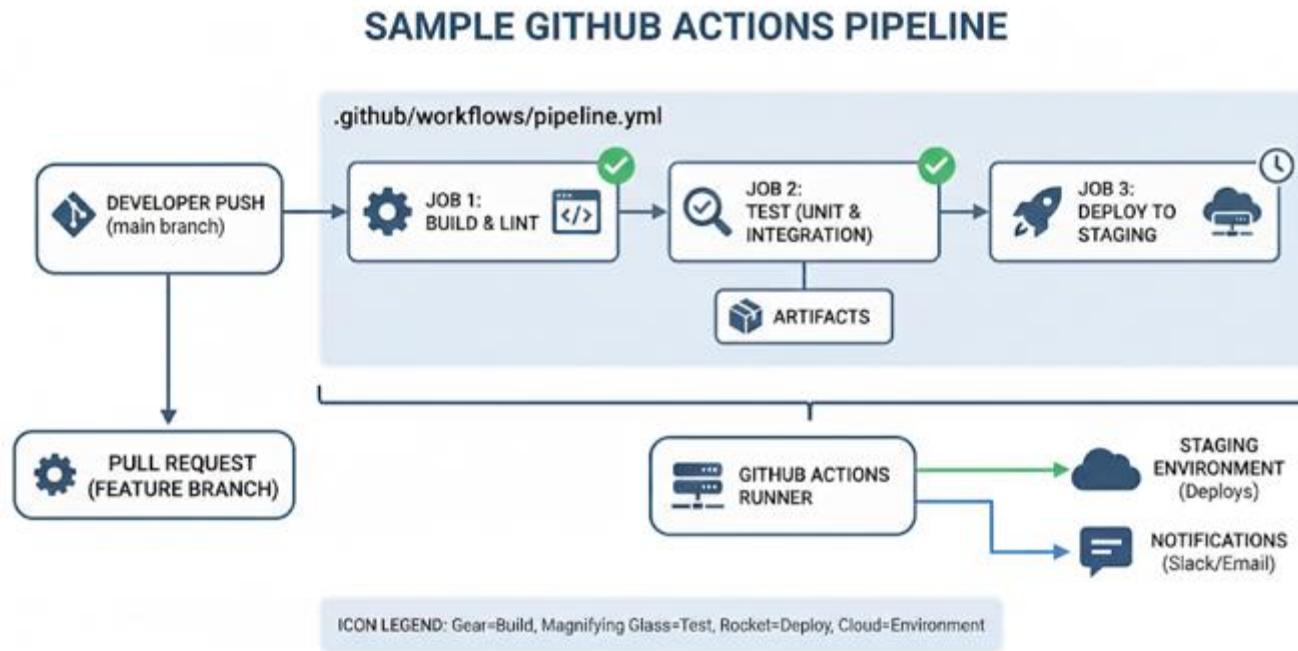
Approach:  
Automate what we can  
Review what we cannot

# Continuous Integration:

Catch mistakes before you push your code!



# Example CI Pipeline



# Developers say:

CI helps us catch bugs earlier

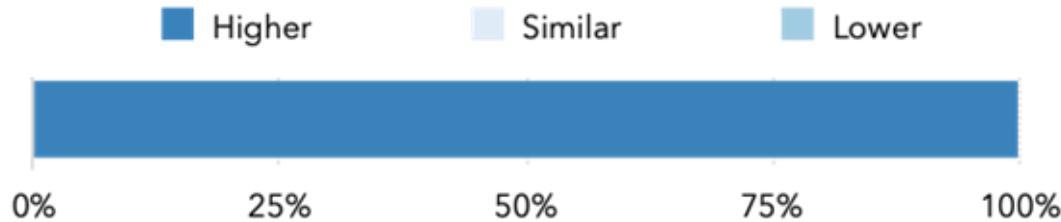
CI makes us less worried about breaking our builds

CI lets us spend less time debugging

"[CI] does have a pretty big impact on [catching bugs]. It allows us to find issues even before they get into our main repo, ... rather than letting bugs go unnoticed, for months, and letting users catch them."

# Developers report:

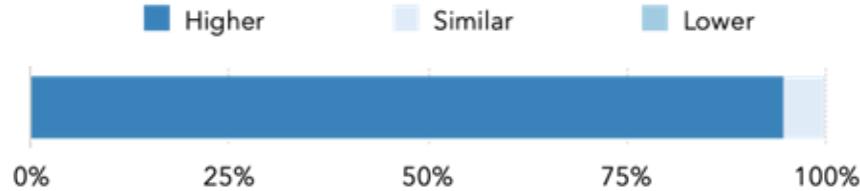
Do developers on projects with CI give (more/similar/less) value to automated tests?



# Developers report:

Do developers on projects with CI give (more/similar/less) value to automated tests?

Do projects with CI have (higher/similar/lower) test quality?

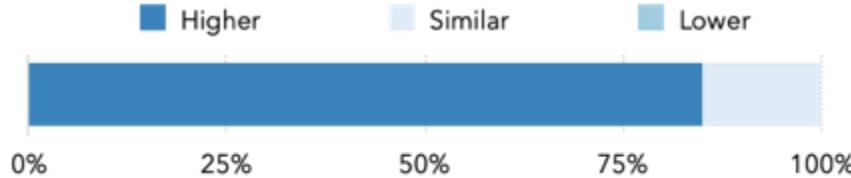


# Developers report:

Do developers on projects with CI give (more/similar/less) value to automated tests?

Do projects with CI have (higher/similar/lower) test quality?

Do projects with CI have (higher/similar/lower) code quality?



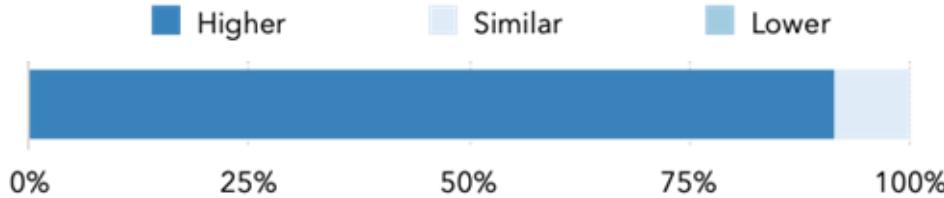
# Developers report:

Do developers on projects with CI give (more/similar/less) value to automated tests?

Do projects with CI have (higher/similar/lower) test quality?

Do projects with CI have (higher/similar/lower) code quality?

Are developers on projects with CI (more/similar/less) productive?

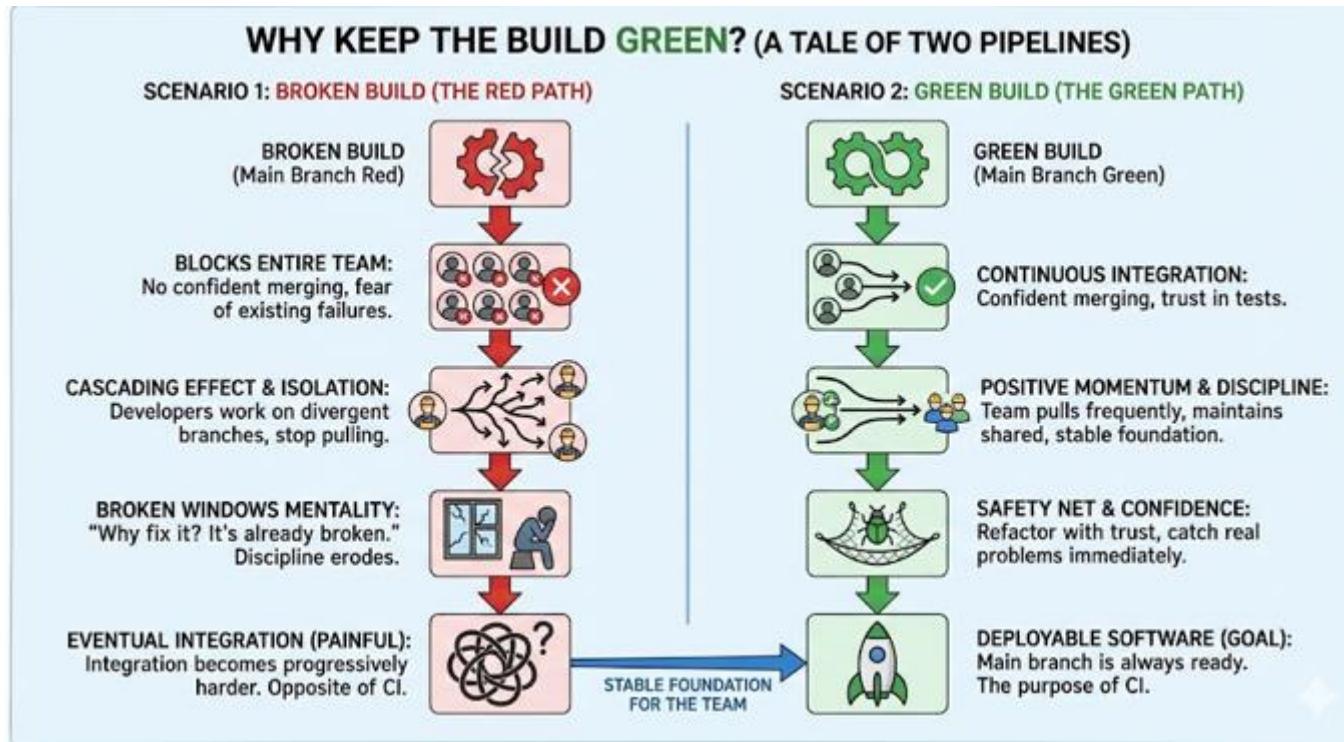


# Observation

CI helps us catch errors  
before others see them  
(but only if we keep it green)

46

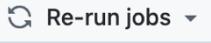
# Why keep the build green?



# NOTE: Update on CI usage this semester

Status:	Number of Teams:
Merged PR's already, CI is green:	10
Merged PR's, CI is red:	2
Have not merged a PR yet:	8

# What can we do to keep build from breaking?

- Keep PRs passing CI
- Merge from main before accepting PR
- Keep an eye on flaky tests:
  - Rerun? 
  - Delete flaky tests?
- Fix CI BEFORE merging new PRs
- Check CI status during code review



# Good

<input type="checkbox"/> 4 Open ✓ 3 Closed	Author ▾ Label ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾
<input type="checkbox"/> <a href="#">Add Answered Support for Topics ✓</a> #20 by Kendric285 was merged 3 days ago	<a href="#">Comment 2</a>
<input type="checkbox"/> <a href="#">Lab 1 ✓</a> #18 by Kendric285 was merged 4 days ago	<a href="#">Comment 2</a>
<input type="checkbox"/> <a href="#">Reduced onTopicsLoaded function complexity ✓</a> #17 by cnk2024 was merged 2 weeks ago	
<input type="checkbox"/> 5 Open ✓ 0 Closed	Author ▾ Label ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾
<input type="checkbox"/> <a href="#">Adding HTML and CSS for Button ✗</a> #36 opened 12 hours ago by Isharma-21	<a href="#">Comment 1</a>
<input type="checkbox"/> 5 Open ✓ 1 Closed	Author ▾ Label ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾
<input type="checkbox"/> <a href="#">Reduced parameters of translateEventArgs, patched all dependencies/callers ✗</a> #22 by IAmCheese1231 was closed last week	

# Bad



PRO Member of CMU-313/f2  
Member of CMU-313

Add missing metadata ✓

- Deleted extra unnecessary files** ✗  
#29 by [REDACTED] was merged 16 hours ago
- Refactorgroups and US3** ✗  
#28 by [REDACTED] was merged 18 hours ago
- Add REST API endpoints for topic resolution and category filtering** ✓  
#12 by [REDACTED] was merged 2 days ago
- Feature/topic resolution impl** ✗  
#11 by [REDACTED] merged 3 days ago
- Feature/answer status system** ✓  
#10 by [REDACTED] merged 4 days ago
- Feature/answer status system** ✓  
#4 by [REDACTED] merged 4 days ago

# CI can run static and dynamic analysis

The screenshot shows a GitHub pull request merge interface. At the top, there's a green button labeled "Merge pull request". Below it, a message says "You can also [open this in GitHub Desktop](#) or view [command line instructions](#)". The main area displays a list of successful CI checks:

- Homework 1 Check / Homework 1 (ubuntu-latest, 16) (pull\_request) Successful in 1m Details
- Lint / Lint (ubuntu-latest, 16) (pull\_request) Successful in 3m Details
- Test / Test (ubuntu-latest, 16, mongo-dev) (pull\_request) Successful in 6m Details
- Test / Test (ubuntu-latest, 16, mongo) (pull\_request) Successful in 5m Details
- Test / Test (ubuntu-latest, 16, redis) (pull\_request) Successful in 5m Details
- Test / Test (ubuntu-latest, 16, postgres) (pull\_request) Successful in 6m Details

Below the checks, a section titled "This branch has no conflicts with the base branch" states "Merging can be performed automatically."

# Static Validation

- Style guides
- Compiler warnings and errors
- Static analysis
  - [FindBugs](#)
  - [clang-tidy](#)
  - [Pylons Webtest](#)
- Code review

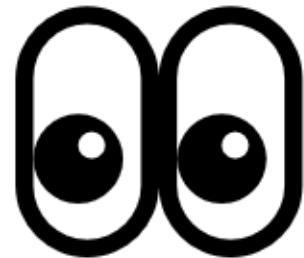
# Style Guide

- List of environment-specific preferred practices
- Could include:
  - Libraries / idioms to use
  - Formatting

# Style Guide Examples

- <https://www.python.org/dev/peps/pep-0008/>
- <https://github.com/airbnb/javascript>
- <https://subversion.apache.org/docs/community-guide/conventions.html>
- <https://google.github.io/styleguide/cppguide.html>
- <https://google.github.io/styleguide/pyguide.html>
- [Linux kernel style guide](#)

# Who writes these style guides?



# Who writes these style guides?

(ad hoc)  Self-proclaimed code protectors

(wisdom) Team veteran developers

(copy-paste) Google search for blog posts by experts

(empirical study) Evidence-based analysis of code styles that correlate with bugs

For problems we can't easily  
automate, we can perform  
code review

# Boeing Model 299 test on October 30, 1935.

- Plane crashed because of locked elevator control surface (opposite effect of MCAS)
- 4 engines were deemed “too complex”
- Test pilots developed checklists to help them fly



# Checklists help manage complex processes



The Checklist: <https://www.newyorker.com/magazine/2007/12/10/the-checklist>

# Dr. Peter Pronovost

- Inspired by B-17 Story
- After checklist, ten-day line-infection rate went from eleven per cent to zero
- In 15 months, only two line infections occurred
- For one hospital, the checklist had prevented forty-three infections and eight deaths, and saved \$2M



## The Pronovost Checklist

Central venous catheters, or lines, are used for medications, blood, fluids or nutrition and can stay in for days or weeks. But bacteria can grow in the line and spread a type of infection to the bloodstream, which causes death in one out five patients who contract it. This five-step checklist for doctors and nurses to use before inserting a line can prevent infections and death.

1. Wash hands with soap and water or an alcohol cleanser

2. Wear sterile clothing—a mask, gloves, and hair covering—and cover patient with a sterile drape, except for a very small hole where the line goes in

3. Clean patient's skin with chlorhexidine (a type of soap) when the line is put in

4. Avoid veins in arm and leg, which are more likely to get infected than veins in chest

5. Check the line for infection each day and remove when no longer needed

Source: Dr. Peter Pronovost

<https://www.wsj.com/articles/SB10001424052748704364004576131963185893084>

# Difference between Pilot and Doctor error?

Which is Developer error more like?

# How to create a checklist?

- Start with problems we have seen before
  - “Safety regulations are written in blood”
- Justify why this is not automatable
- Not all checklist items need to be very specific
  - An item could be “does this team know we are proposing this change”

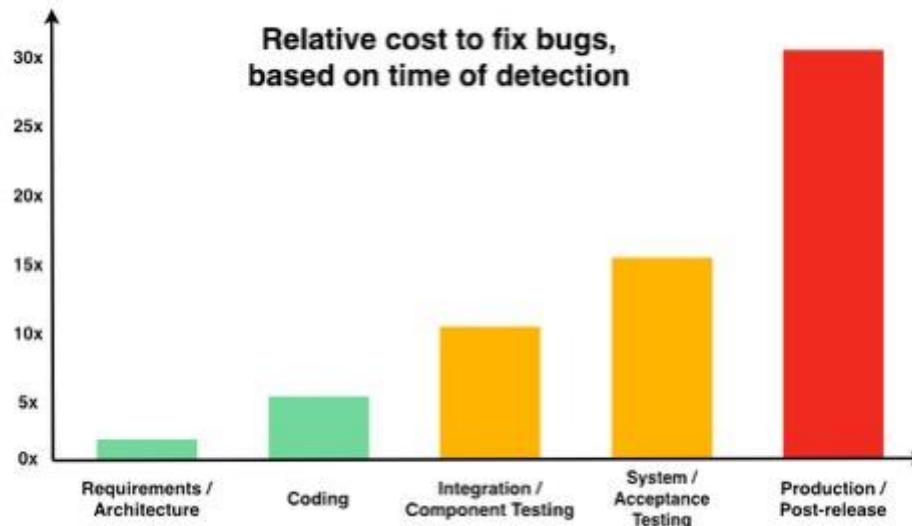
# Activity: Create a checklist

- In pairs, think about dumb mistakes your “friend” made the last time they were coding.
  - Write your names on a piece of paper.
  - Write down two checklist items that would have caught those errors.
- Divide into teams: left and right sides of the classroom.
- Which team had the most unique/good entries in their list?

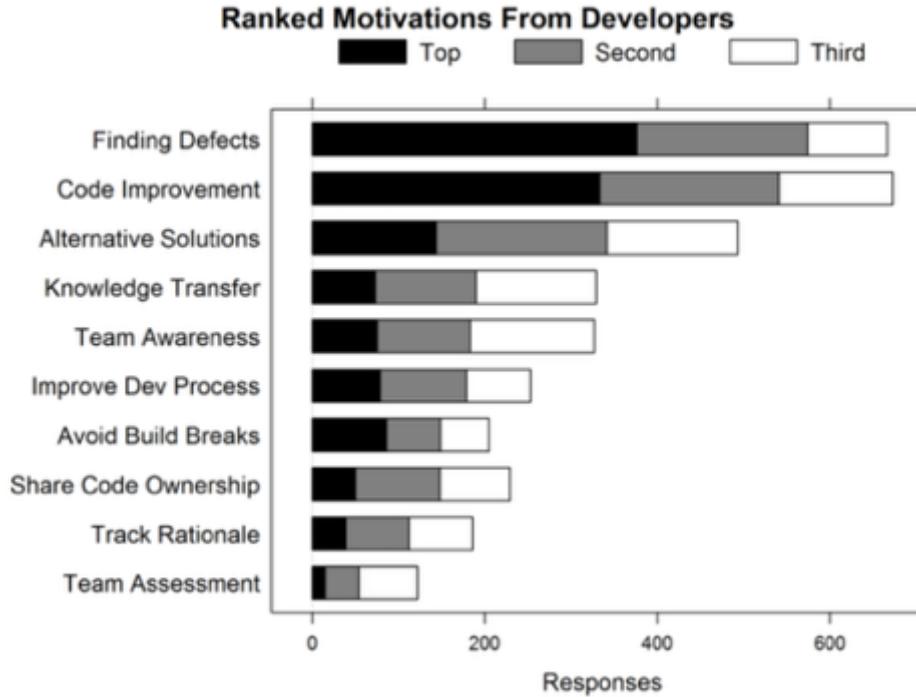
# Expectations and Outcomes for code review

# Motivation

- Linus's Law: “Given enough eyeballs, all bugs are shallow.”
  - - The Cathedral and the Bazaar, Eric Raymond

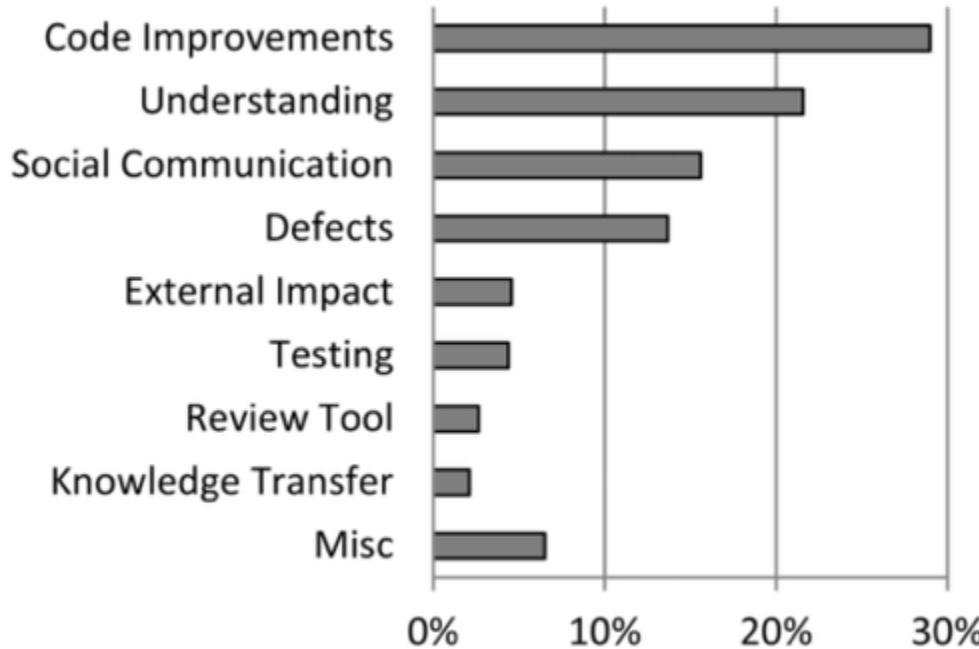


# Code Review at Microsoft



Bacchelli, Alberto and Christian Bird. "Expectations, outcomes, and challenges of modern code review." Proceedings of the 2013 International Conference on Software Engineering. IEEE Press, 2013.

# Outcomes (Analyzing Reviews)



# Mismatch of Expectations and Outcomes

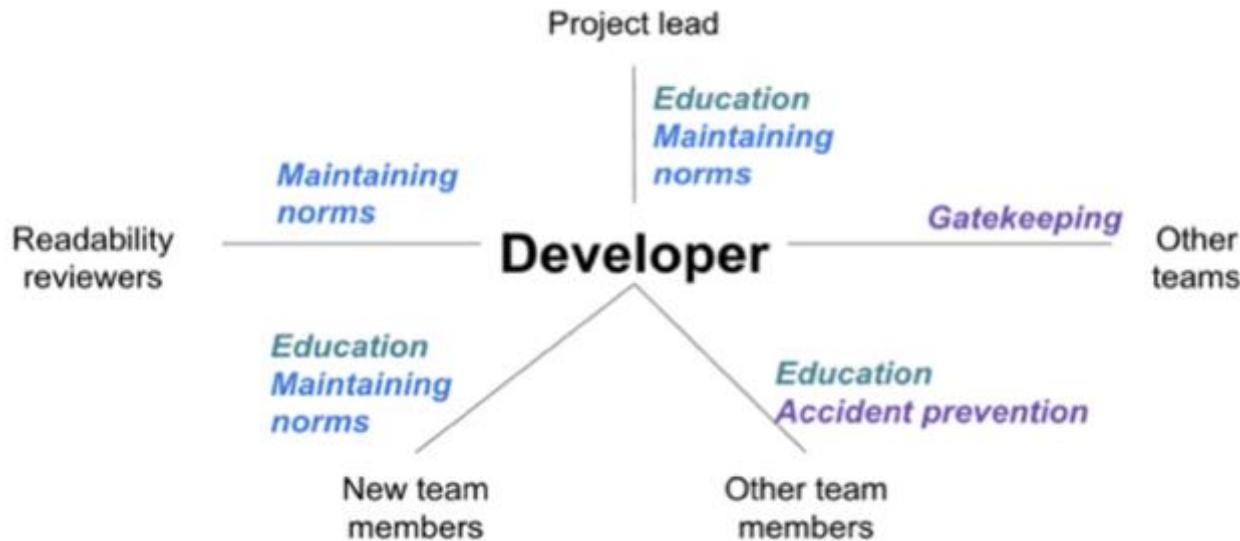
- Low quality of code reviews
  - Reviewers look for easy errors, as formatting issues
  - Miss serious errors
- Understanding is the main challenge
  - Understanding the reason for a change
  - Understanding the code and its context
  - Feedback channels to ask questions often needed
- No quality assurance on the outcome

# Code Review at Google

- Introduced to “force developers to write code that other developers could understand”
- Three benefits:
  - checking the consistency of style and design
  - ensuring adequate tests
  - improving security by making sure no single developer could commit arbitrary code without oversight

Caitlin Sadowski, Emma Söderberg, Luke Church, Michal Sipko, and Alberto Bacchelli. 2018. Modern Code Review: A Case Study at Google. International Conference on Software Engineering

# Reviewing Relationships



# The State of Code Review survey



# Code Review

- Start with the “big ideas”
- Automate the little things
- Focus on understanding
- Remember a person wrote the code
- Don’t overwhelm the person with feedback

# Don't forget that coders are people with feelings

- A coder's self-worth is in their artifacts
- CI can avoid embarrassment
- Identify defects, not alternatives; do not criticize coder
  - "you didn't initialize variable a" -> "I don't see where variable a is initialized"
- Avoid defending code; avoid discussions of solutions/alternatives
- Reviewers should not "show off" that they are better/smarter
- Avoid style discussions if there are no guidelines
- The coder gets to decide how to resolve fault

# Risk Analysis:

- Probability a human makes a mistake: Very Likely
- Severity: ranges, but could be extensive

Solution:

Use CI to catch your mistakes, make you look better, and mitigate your risks!

Use Code review to teach and learn

