WE'LL NEED A RISK ANALYSIS ON THIS PROJECT BEFORE I CAN APPROVE IT.
RISK 1: INDECISIVENESS
RISK 2: OVERANALYSIS
RISK 3: CLUELESSNESS
RISK 4: MICROMANAGE-MENT...
CLICK CLICK CLICK
I DON'T UNDERSTAND THESE RISKS.
THAT'S NUMBER THIRTY-SIX.
www.unitedmedia.com
S. Adams
© 1997 United Feature Syndicate, Inc.
11/8/97

Build Software Safely!

17-313 Fall 2025

Foundations of Software Engineering

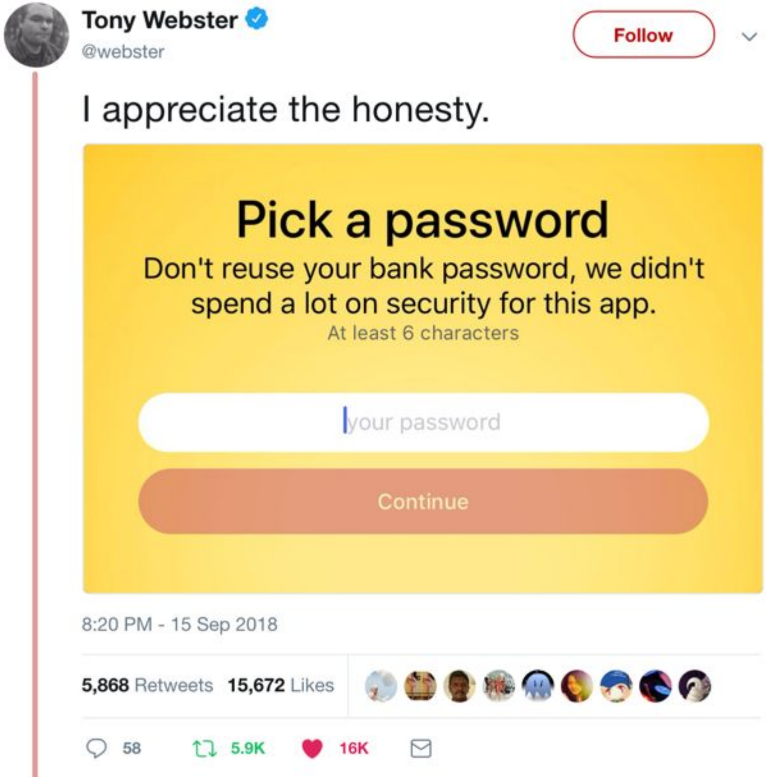
https://cmu-17313q.github.io

Eduardo Feo Flushing

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

Risk



Tony Webster
@webster

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

8:20 PM - 15 Sep 2018

5,868 Retweets 15,672 Likes

Definition: Risk

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

Conrow, E. 2003. Effective Risk Management: Some Keys to Success, 2nd ed. Reston, VA, USA: American Institute of Aeronautics and Astronautics (AIAA).

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 outcome

Conrow, E. 2003. Effective Risk Management: Some Keys to Success, 2nd ed. Reston, VA, USA: American Institute of Aeronautics and Astronautics (AIAA).

Internal vs. External Risk

Risks that we can control

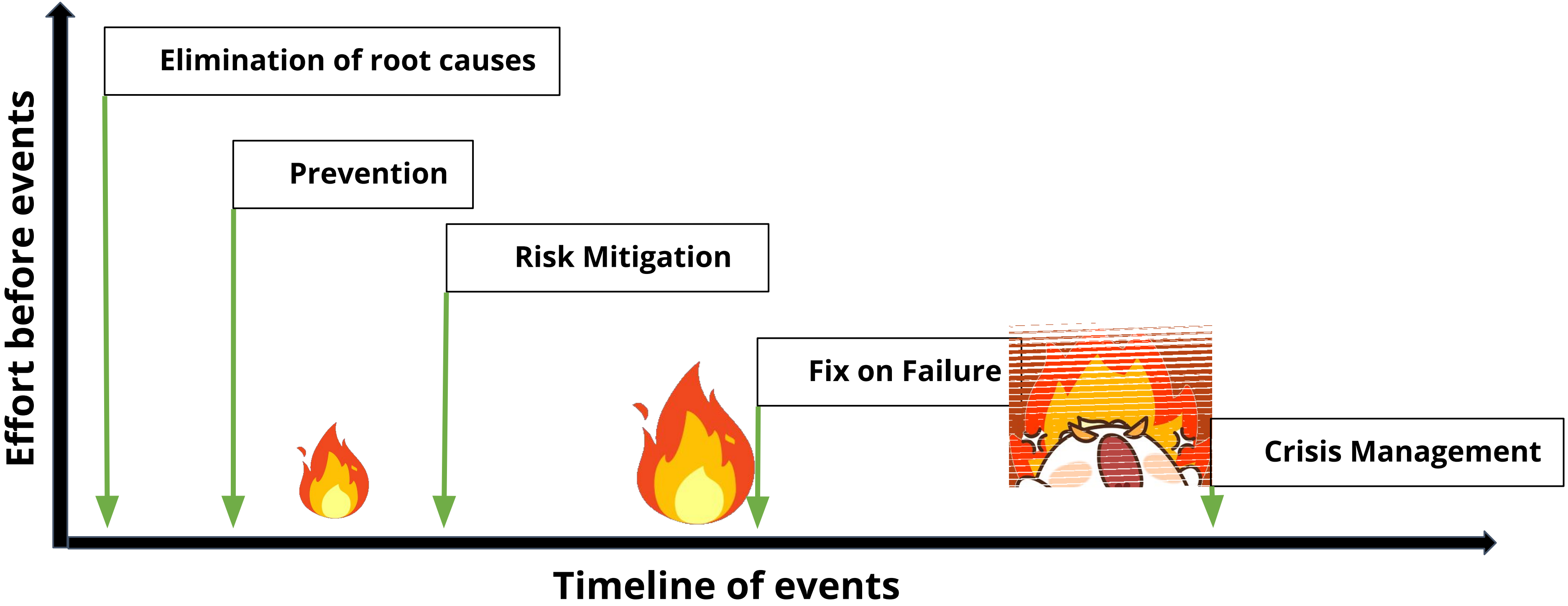
Risks that we cannot control

Levels of Risk Management

1. **Elimination of root causes:**
   - Identify and eliminate factors that make it possible for risks to exist at all.
2. **Prevention:**
   - Implement and execute a plan as part of the software project to identify risks and prevent them from becoming problems.
3. **Risk mitigation:**
   - Plan ahead of time to provide resources to cover risks if they occur, but you don't reduce the chance of the risk happening.
4. **Fix on failure:**
   - Detect and react to risks quickly, but only after they have occurred.
5. **Crisis management:**
   - Fire fighting; address risks only after they become problems.

"Rapid Development: Taming Wild Software Schedules," Steve McConnell, 1996

Levels of Risk Management



Elimination of root causes

Prevention

Risk Mitigation

Fix on Failure

Crisis Management

Effort before events

Timeline of events

“Rapid Development: Taming Wild Software Schedules,” Steve McConnell, 1996

Levels of Risk Management

1. **Elimination of root causes:**
   - You build the house with fireproof materials and remove all potential fire hazards to prevent the fire from ever occurring.
2. **Prevention**
   - You install smoke detectors, inspect wiring, and remove fire hazards to reduce the chance of a fire starting.
3. **Risk mitigation**
   - You install fire extinguishers and sprinklers to reduce the damage when a fire occurs but take no steps to prevent the fire.
4. **Fix on failure**
   - You have smoke detectors that alert you to the fire, and you react quickly once it's detected.
5. **Crisis management**
   - You wait until the fire is visible and then call the fire department to put it out.

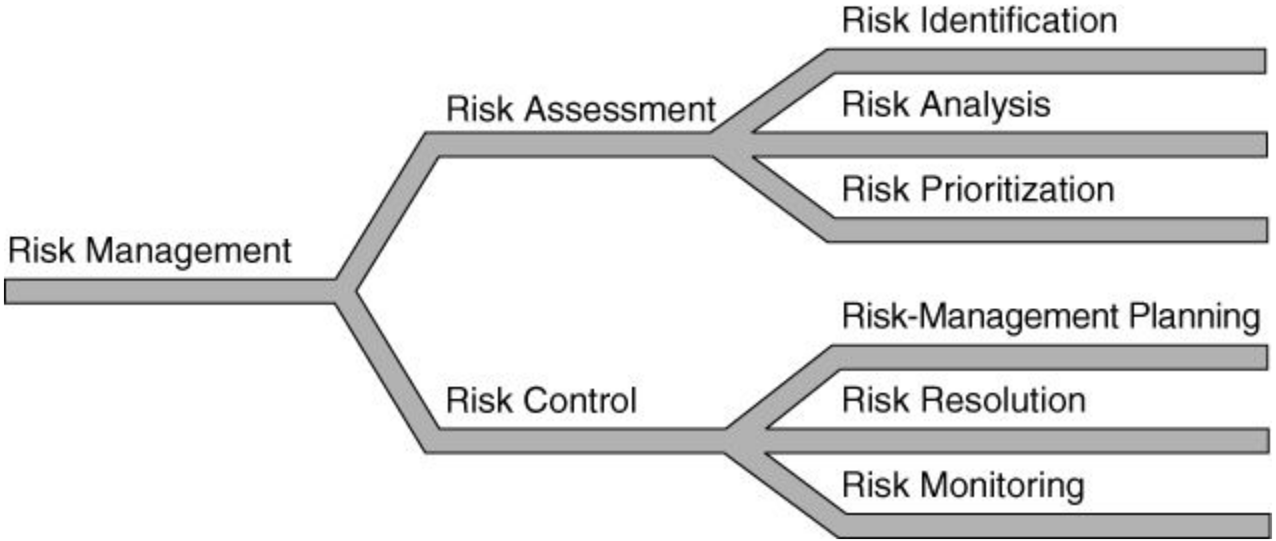
Levels of Risk Management

1. **Elimination of root causes:**
   - Identify and eliminate factors that make it possible for risks to exist at all.
2. **Prevention:**
   - Implement and execute a plan as part of the software project to identify risks and prevent them from becoming problems.
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. **Fix on failure:**
   - Detect and react to risks quickly, but only after they have occurred.
5. **Crisis management:**
   - Fire fighting; address risks only after they become problems.

Not covered in this class

"Rapid Development: Taming Wild Software Schedules," Steve McConnell, 1996

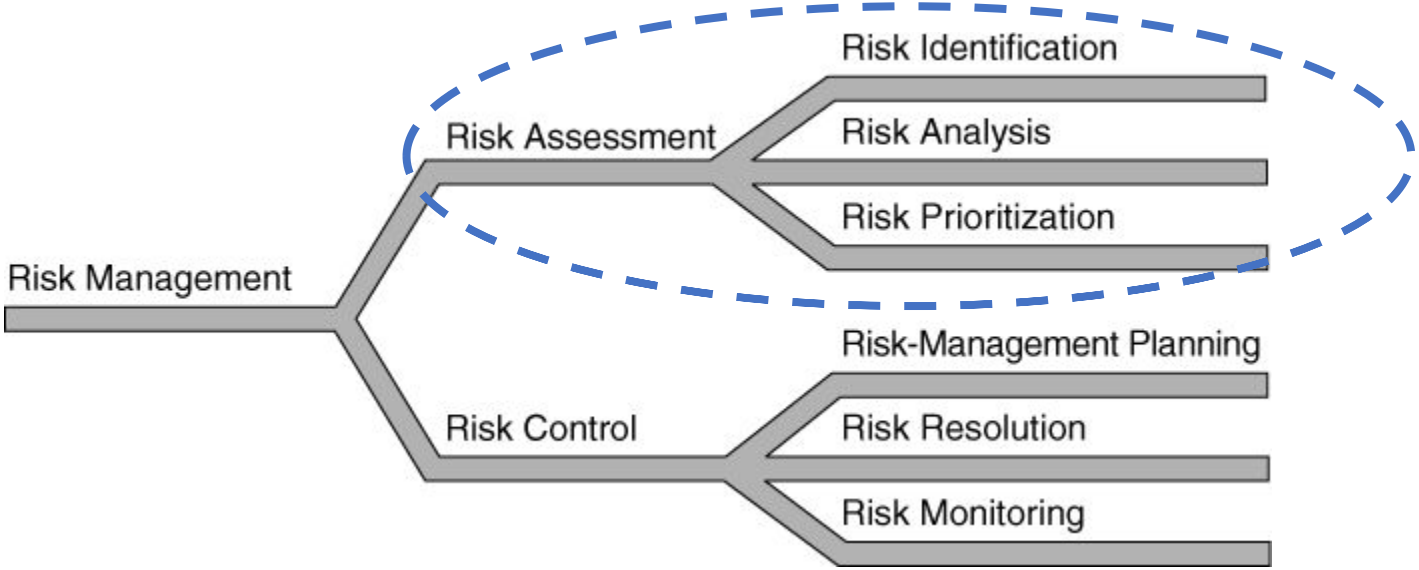
Risk Management



These are core tasks that support prevention, mitigation, and root-cause elimination

“Rapid Development: Taming Wild Software Schedules,” Steve McConnell, 1996

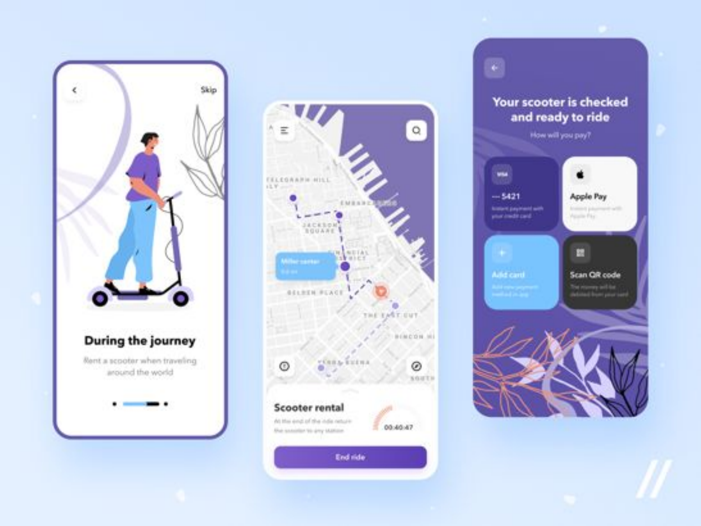
Risk Management



"Rapid Development: Taming Wild Software Schedules," Steve McConnell, 1996

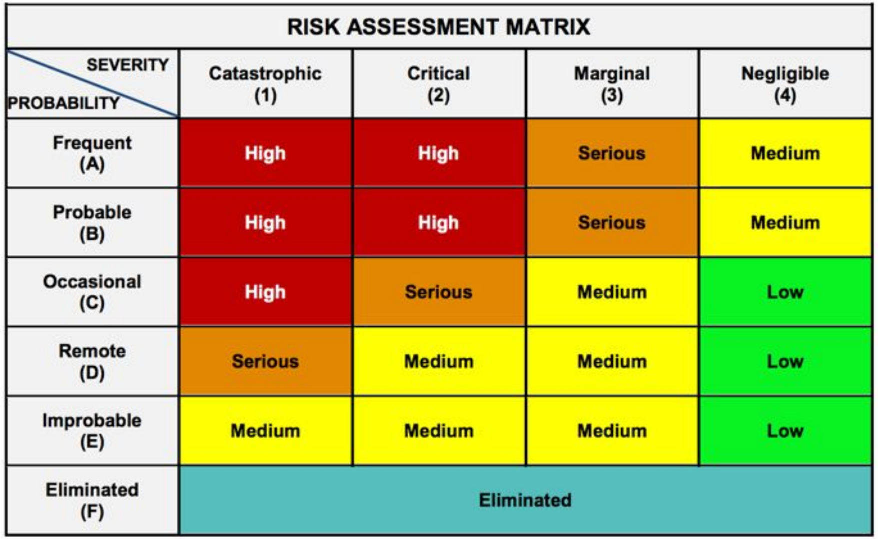
Team Exercise: Risk Identification

- What risks exist for the scooter app?



Risk assessment matrix

TABLE III. Risk assessment matrix



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

DO-178b, Software Considerations in Airborne Systems and Equipment Certification, RTCA, 1992

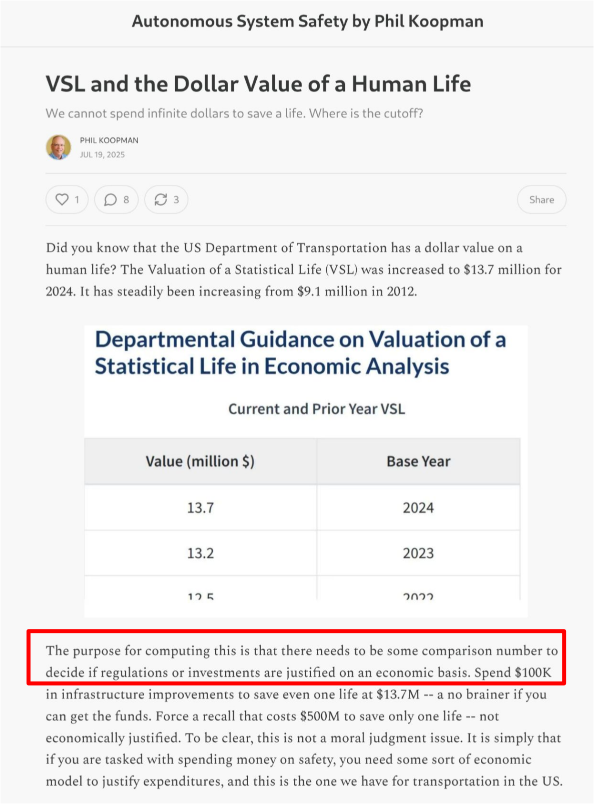
Risk Analysis

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
...	...	...	...
Total			12

Sad truth:

Risk analysis often becomes a numbers game to justify regulations or investments, rather than a tool for genuine safety improvement.

The purpose for computing this is that there needs to be some comparison number to decide if regulations or investments are justified on an economic basis.



Autonomous System Safety by Phil Koopman

VSL and the Dollar Value of a Human Life

We cannot spend infinite dollars to save a life. Where is the cutoff?

PHIL KOOPMAN
JUL 19, 2025

Did you know that the US Department of Transportation has a dollar value on a human life? The Valuation of a Statistical Life (VSL) was increased to $13.7 million for 2024. It has steadily been increasing from $9.1 million in 2012.

Departmental Guidance on Valuation of a Statistical Life in Economic Analysis

Current and Prior Year VSL

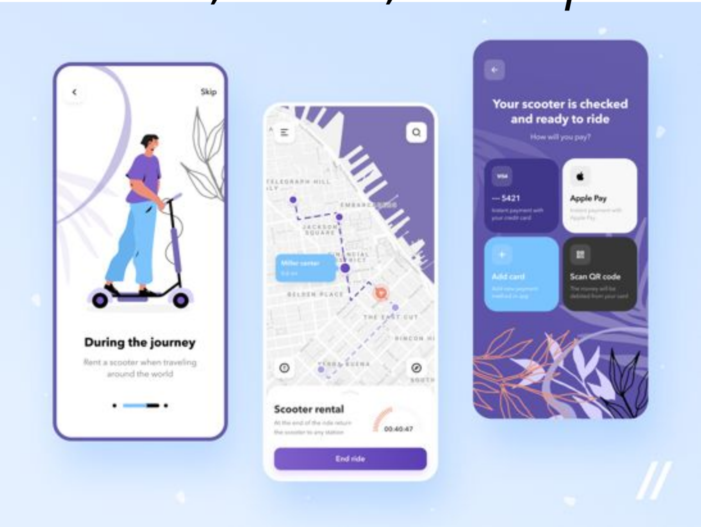
Value (million $)	Base Year
13.7	2024
13.2	2023
12.5	2022

The purpose for computing this is that there needs to be some comparison number to decide if regulations or investments are justified on an economic basis. Spend $100K in infrastructure improvements to save even one life at $13.7M -- a no brainer if you can get the funds. Force a recall that costs $500M to save only one life -- not economically justified. To be clear, this is not a moral judgment issue. It is simply that if you are tasked with spending money on safety, you need some sort of economic model to justify expenditures, and this is the one we have for transportation in the US.

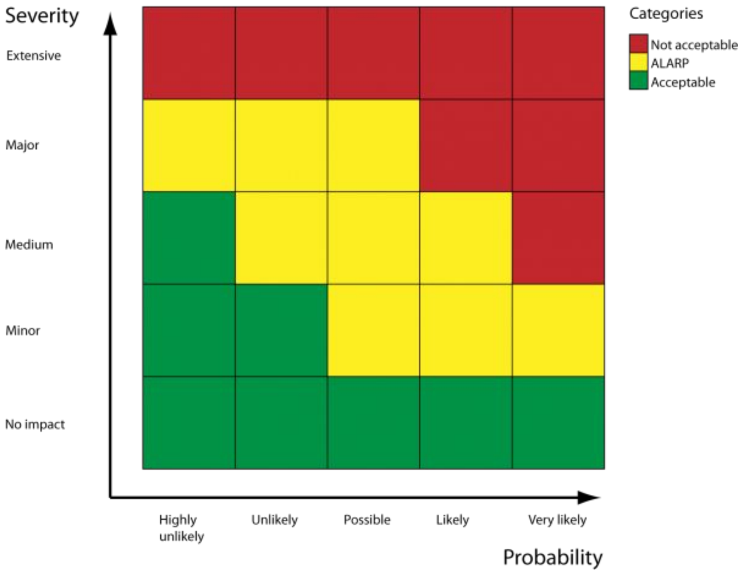
Exercise: Risk Analysis

- What is the risk probability and severity for your scooter app?

  Frequent, Probable, Not so often, almost never
  Extensive, Major, Medium, Minor, No Impact


Skip
During the journey
Rent a scooter when traveling around the world
TELEGRAPH HILL
EMBARCADERO
JACKSON SQUARE
FINANCIAL DISTRICT
Miller center
BELDEN PLACE
THE EAST CUT
RINCON HILL
YERBA BUENA
SOUTH
Scooter rental
At the end of the ride return the scooter to any station
00:40:47
End ride
Your scooter is checked and ready to ride
How will you pay?
VISA
···· 5421
Instant payment with your credit card
Apple Pay
Instant payment with Apple Pay
Add card
Add new payment method in app
Scan QR code
The money will be debited from your card

Risk Prioritization
Focus on risks with the highest exposure

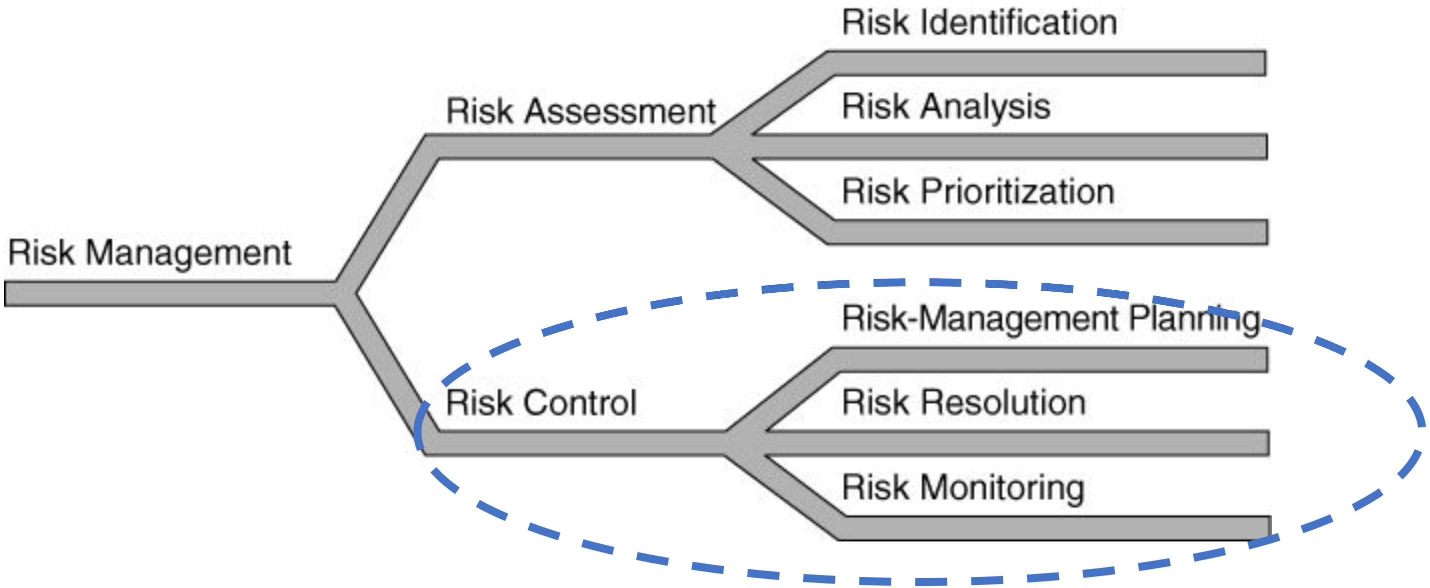


Severity	Highly unlikely	Unlikely	Possible	Likely	Very likely
Extensive	🟥	🟥	🟥	🟥	🟥
Major	🟨	🟨	🟨	🟥	🟥
Medium	🟩	🟨	🟨	🟨	🟥
Minor	🟩	🟩	🟨	🟨	🟨
No impact	🟩	🟩	🟩	🟩	🟩

Categories

- 🟥 Not acceptable
- 🟨 ALARP
- 🟩 Acceptable

Risk Management



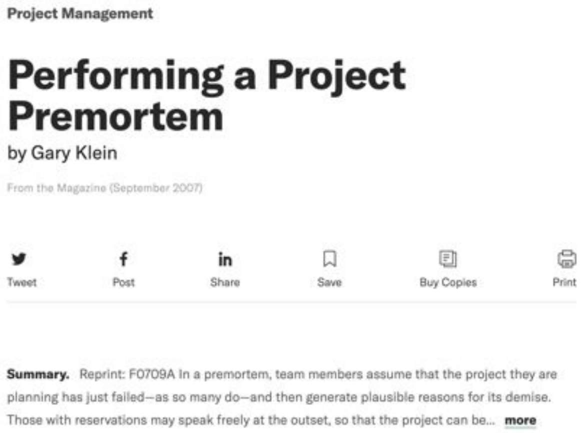
“Rapid Development: Taming Wild Software Schedules,” Steve McConnell, 1996

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?
- Incorporate risks into your schedule
  - Don’t assume everything will go smoothly between now and the end of the semester!

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

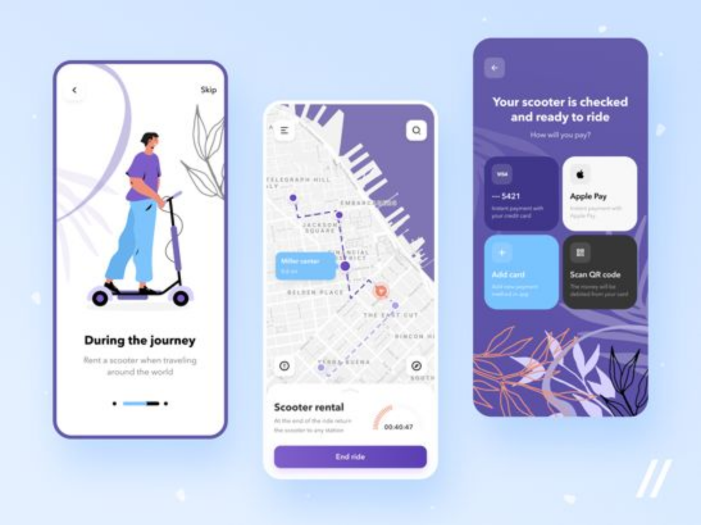
by Gary Klein

From the Magazine (September 2007)

Summary. Reprint: F0709A 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**

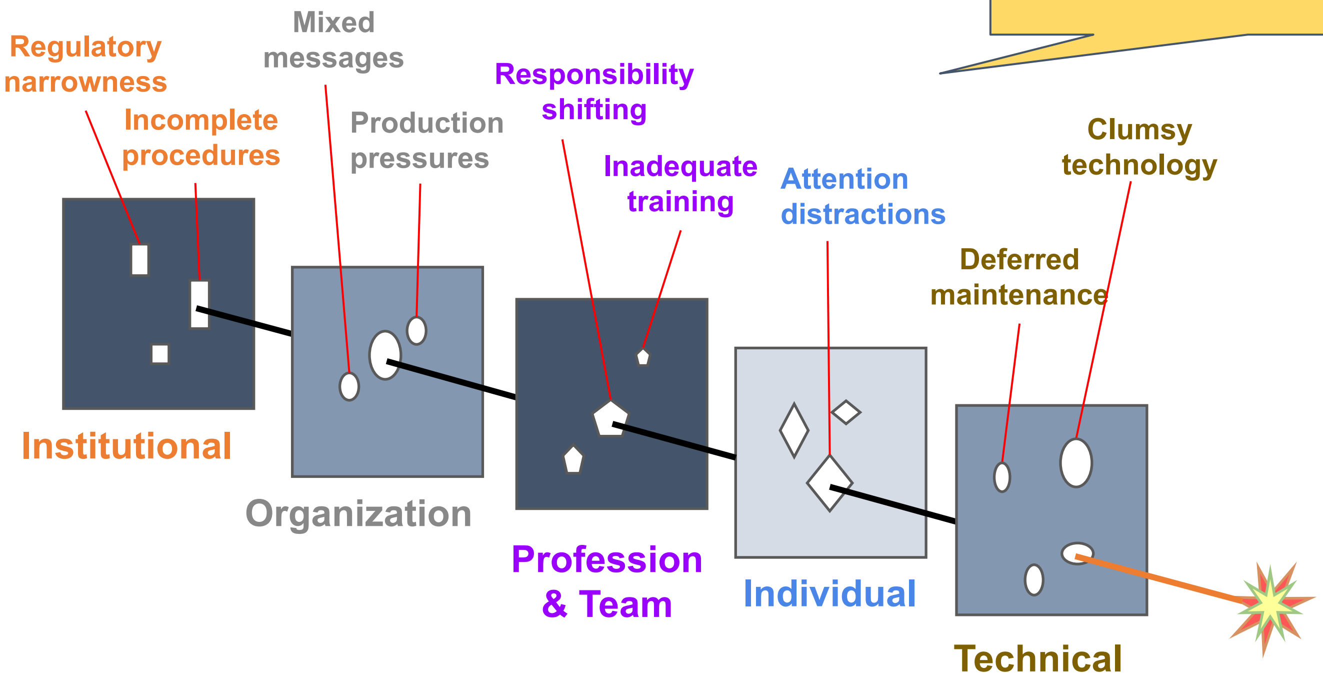
Discussion: Risk Elimination and Mitigation

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



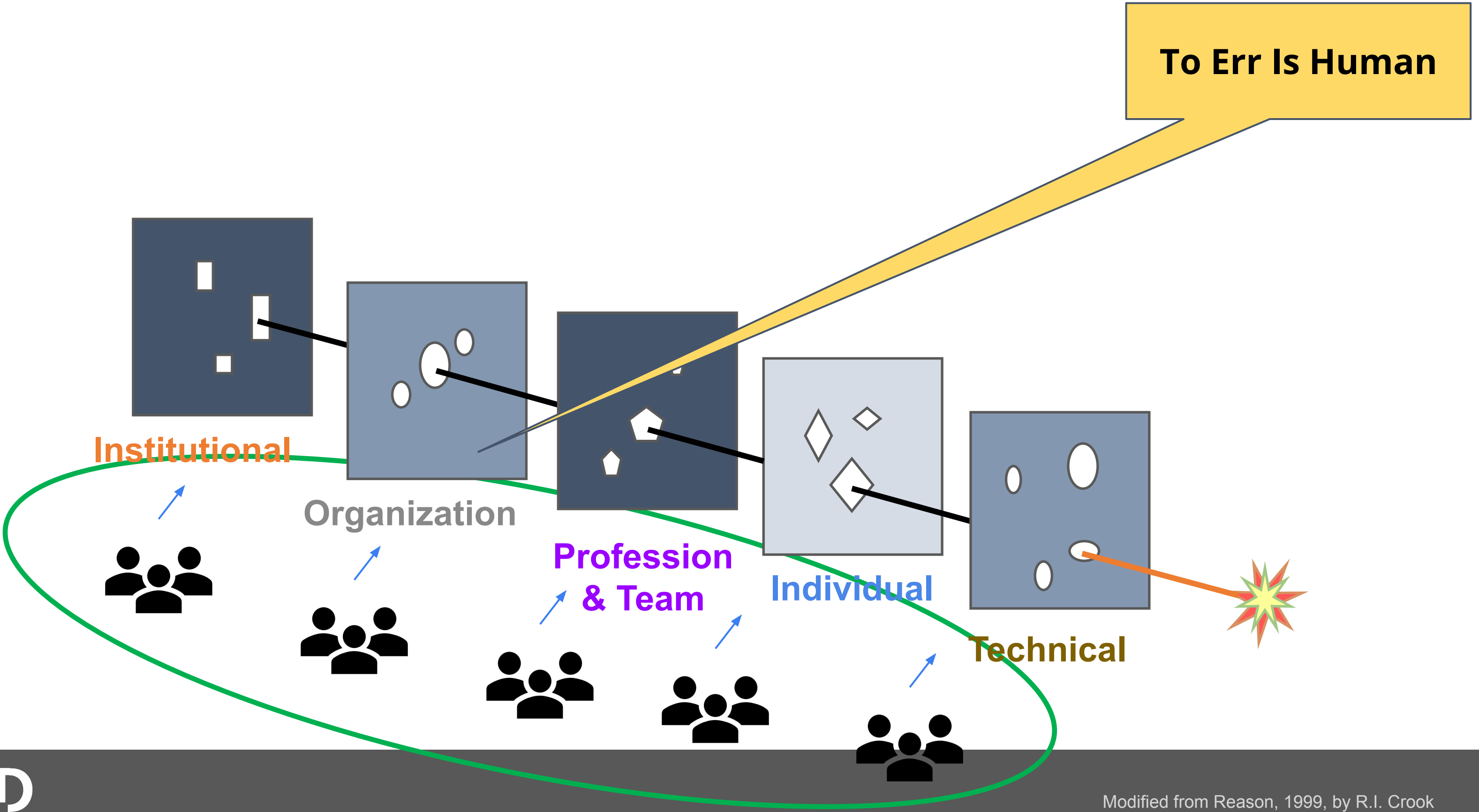
The Swiss cheese model

Risk control needs multiple overlapping defenses



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

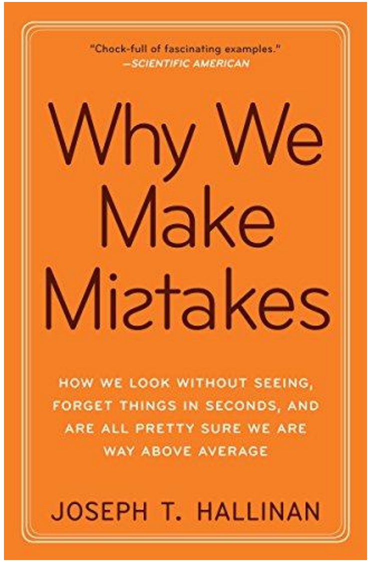
The Swiss cheese model



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

Can we remove human error?

Why do we make mistakes?


"Chock-full of fascinating examples."
—SCIENTIFIC AMERICAN

Why We Make Mistakes

HOW WE LOOK WITHOUT SEEING, FORGET THINGS IN SECONDS, AND ARE ALL PRETTY SURE WE ARE WAY ABOVE AVERAGE

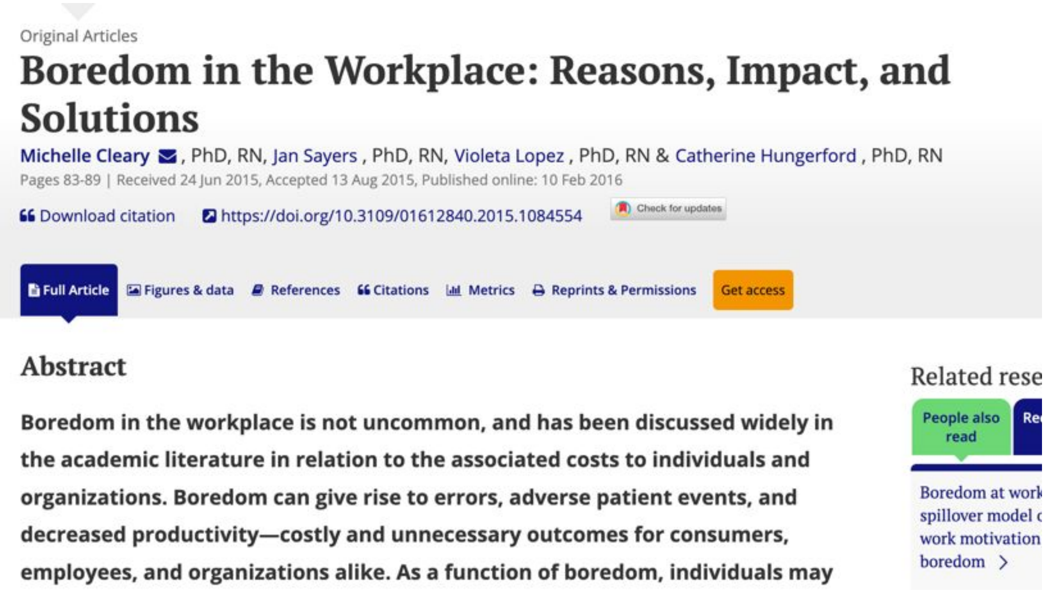
JOSEPH T. HALLINAN

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.

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 result of boredom, individuals may feel overworked or underutilized, 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

Abstract

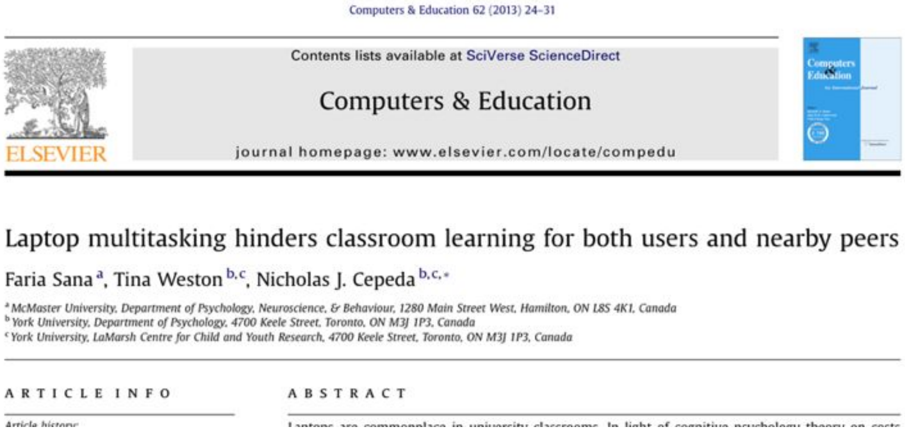
Boredom in the workplace is not uncommon, and has been discussed widely in the academic literature in relation to the associated costs to individuals and organizations. Boredom can give rise to errors, adverse patient events, and decreased productivity—costly and unnecessary outcomes for consumers, employees,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

Related rese

Boredom at work spillover model c work motivation boredom ›

Cognitive Load

- …" students who switch back and forth between attending a lecture and checking email, 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

ELSEVIER

Laptop multitasking hinders classroom learning for both users and nearby peers

Faria Sana[a], Tina Weston[b,c], Nicholas J. Cepeda[b,c,*]

[a] McMaster University, Department of Psychology, Neuroscience, & Behaviour, 1280 Main Street West, Hamilton, ON L8S 4K1, Canada
[b] York University, Department of Psychology, 4700 Keele Street, Toronto, ON M3J 1P3, Canada
[c] York University, LaMarsh Centre for Child and Youth Research, 4700 Keele Street, Toronto, ON M3J 1P3, Canada

ARTICLE INFO

Article history:

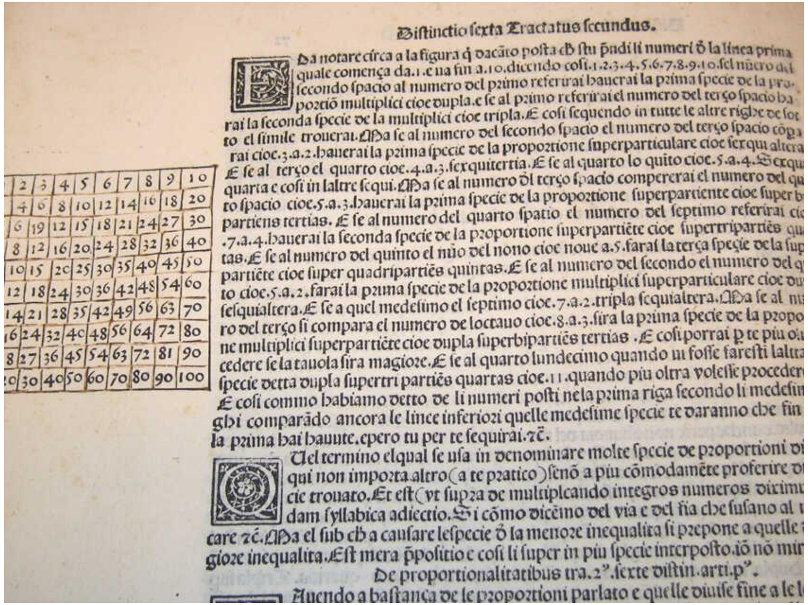
ABSTRACT

Laptops are commonplace in university classrooms. In light of cognitive psychology theory on costs catch

Can we ~~remove~~ human error?

Can we catch human error before releasing our code?

Can we automate tasks to prevent problems?



Diſtinctio ſexta Tractatus ſecundus.

2	3	4	5	6	7	8	9	10
4	6	8	10	12	14	16	18	20
6	19	12	15	18	21	24	27	30
8	12	16	20	24	28	32	36	40
10	15	20	25	30	35	40	45	50
12	18	24	30	36	42	48	54	60
14	21	28	35	42	49	56	63	70
16	24	32	40	48	56	64	72	80
8	27	36	45	54	63	72	81	90
20	30	40	50	60	70	80	90	100

E Da notare circa a la figura q̇ dacāto poſta cħ ſtu p̄ndi li numeri ō la linea prima
quale comença da.1.e ua fin a.10.dicendo coſi.1.2.3.4.5.6.7.8.9.10.ſel nūero del
ſecondo ſpacio al numero del primo referirai haurai la prima ſpecie de la pro-
portiō multiplici cioe dupla.e ſe al primo referirai el numero del terço ſpacio ha
rai la ſeconda ſpecie de la multiplici cioe tripla.E coſi ſequendo in tutte le altre righe de ſor
to el ſimile trouerai.Ma ſe al numero del ſecondo ſpacio el numero del terço ſpacio cōpa
rai cioe.3.a.2.haurai la prima ſpecie de la proportione ſuperparticulare cioe ſexqui altera
E ſe al terço el quarto cioe.4.a.3.ſexquitertia.E ſe al quarto lo quito cioe.5.a.4.Sexqu
quarta e coſi in laltre ſequi.Ma ſe al numero d̄l terço ſpacio compererai el numero del qu
to ſpacio cioe.5.a.3.haurai la prima ſpecie de la proportione ſuperpartiente cioe ſuper b
partiens tertias. E ſe al numero del quarto ſpatio el numero del ſeptimo referirai cio
.7.a.4.haurai la ſeconda ſpecie de la proportione ſuperpartiēte cioe ſupertripartiēs qu
tas.E ſe al numero del quinto el nūo del nono cioe noue a.5.farai la terça ſpegie de la ſup
partiēte cioe ſuper quadripartiēs quintas.E ſe al numero del ſecondo el numero del q
to cioe.5.a.2.farai la prima ſpecie de la proportione multiplici ſuperparticulare cioe du
ſeſquialtera.E ſe a quel medeſimo el ſeptimo cioe.7.a.2.tripla ſequialtera.Ma ſe al nu
ro del terço ſi compara el numero de loctauo cioe.8.a.3.ſira la prima ſpecie de la propo
ne multiplici ſuperpartiēte cioe dupla ſuperbipartiēs tertias .E coſi porrai ꝑ te piu o
cedere ſe la tauola ſira magiore.E ſe al quarto lundecimo quando ui foſſe fareſti laltra
ſpecie detta dupla ſupertri partiēs quartas cioe.11.quando piu oltra voleſſe proceder
E coſi commo habiamo detto de li numeri poſti ne la prima riga ſecondo li medeſim
ghi comparādo ancora le linee inferiori quelle medeſime ſpecie te daranno che fin
la prima hai hauute.e pero tu per te ſequirai.ꝛc̄.

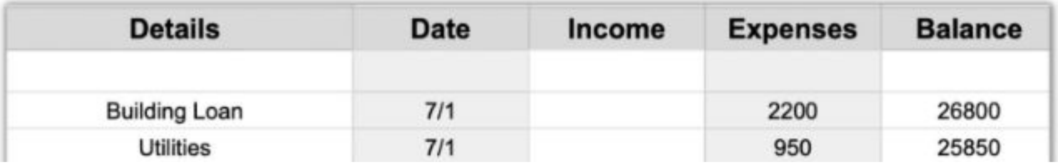
Q Uel termino elqual ſe uſa in denominare molte ſpecie de proportioni d
qui non importa altro(a te pratico)ſenō a piu cōmodamēte proferire d
cie trouato.Et eſt(vt ſupra de multiplcando integros numeros diximu
dam ſyllabica adiectio.Si cōmo dicēmo del via e del fia che ſuſano al
care ꝛc̄.Ma el ſub ch a cauſare leſpecie ō la menore inequalita ſi prepone a quelle
giore inequalita.Eſt mera ꝑpoſitio e coſi li ſuper in piu ſpecie interpoſto.iō nō mi

De proportionalitatibus tra.2°.ſexte diſtin.arti.p°.

Auendo a baſtança de le proportioni parlato e quelle diuiſe fine a le l

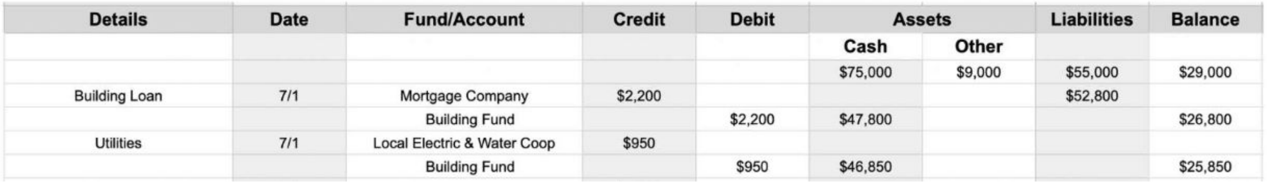
Double entry accounting

SINGLE ENTRY



Details	Date	Income	Expenses	Balance
Building Loan	7/1		2200	26800
Utilities	7/1		950	25850

DOUBLE ENTRY

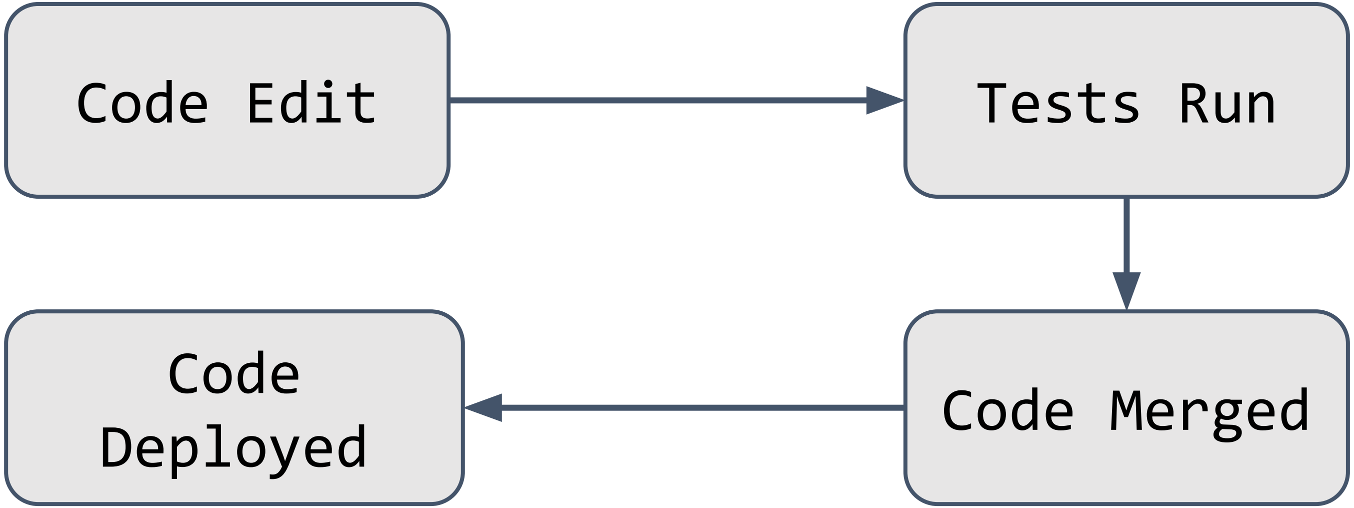


Details	Date	Fund/Account	Credit	Debit	Assets		Liabilities	Balance
					Cash	Other		
					$75,000	$9,000	$55,000	$29,000
Building Loan	7/1	Mortgage Company	$2,200				$52,800	
		Building Fund		$2,200	$47,800			$26,800
Utilities	7/1	Local Electric & Water Coop	$950					
		Building Fund		$950	$46,850			$25,850

Approach:

Automate what we can,
Review what we cannot

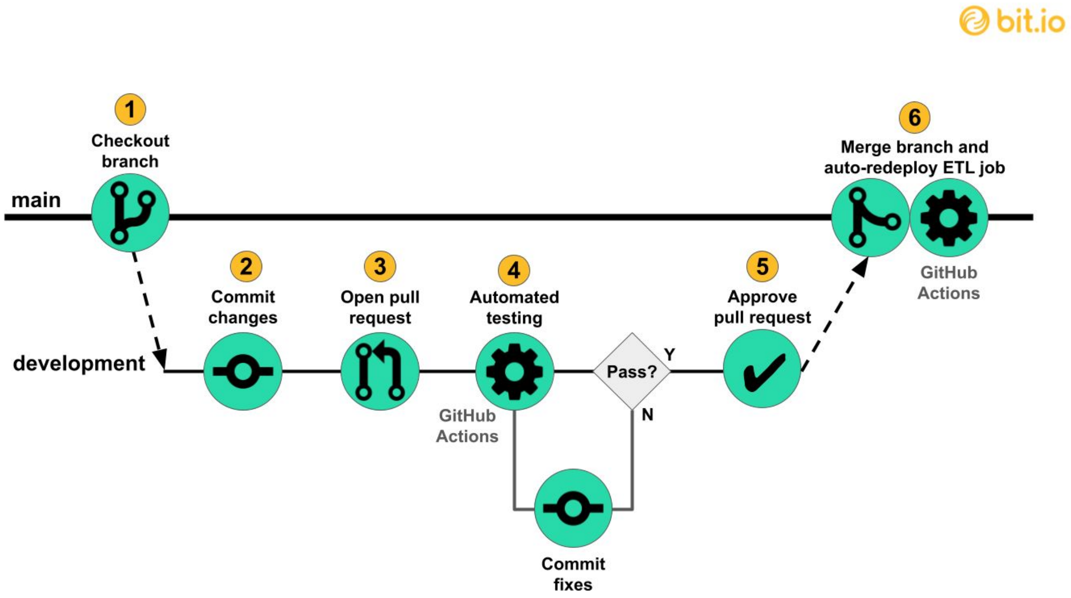
CI/CD Pipeline overview



Continuous Integration:

Catch mistakes before merging your code!
CI/CD **reduces project risk** by catching mistakes early.

Example CI Workflow



Source: https://innerjoin.bit.io/making-a-simple-data-pipeline-part-4-ci-cd-with-github-actions-733251f211a6

Observation

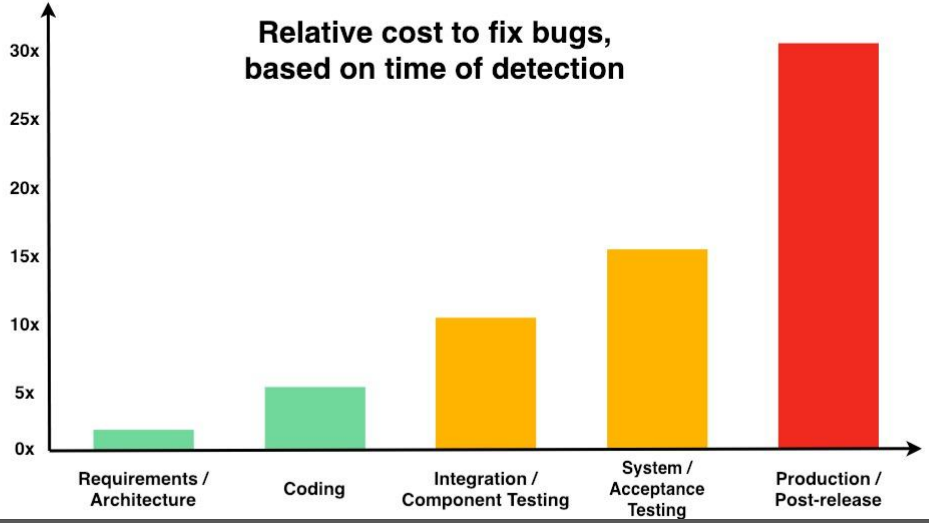
Continuous Integration helps us catch errors before others see them

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

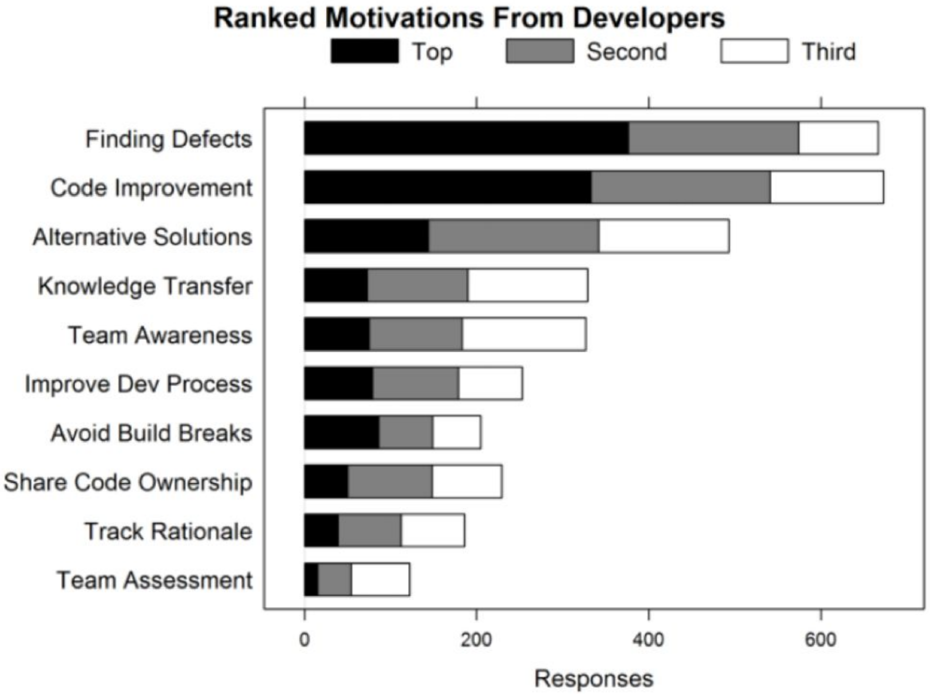
Code reviews reduce risk by catching errors humans introduce, especially those automation can’t detect.

Motivation

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

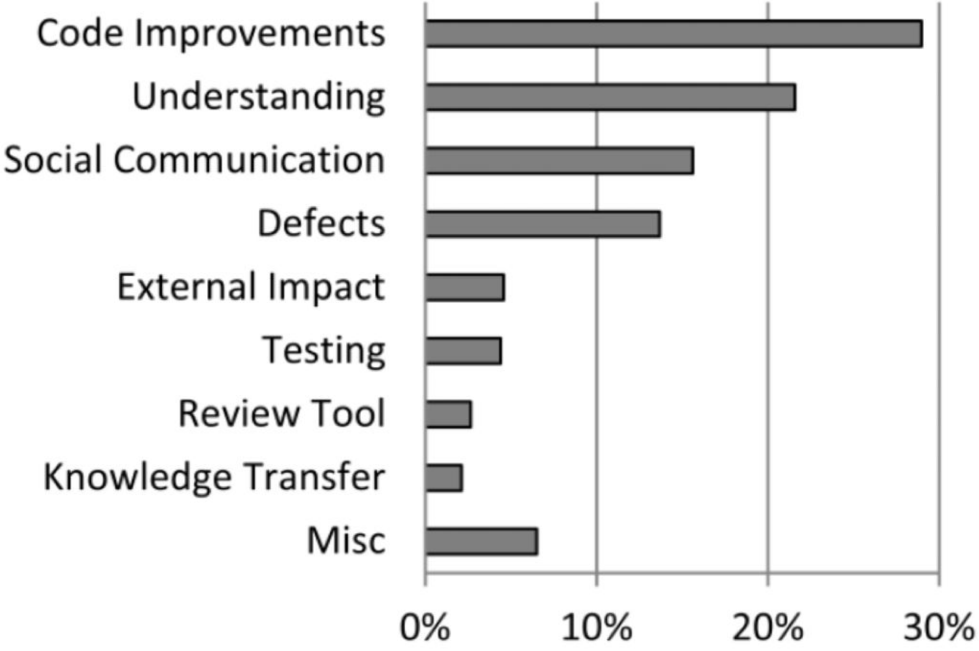


Code Review at Microsoft



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 often focus on easy-to-spot issues, such as formatting, and 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
- There is often no assurance of the review's overall quality

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

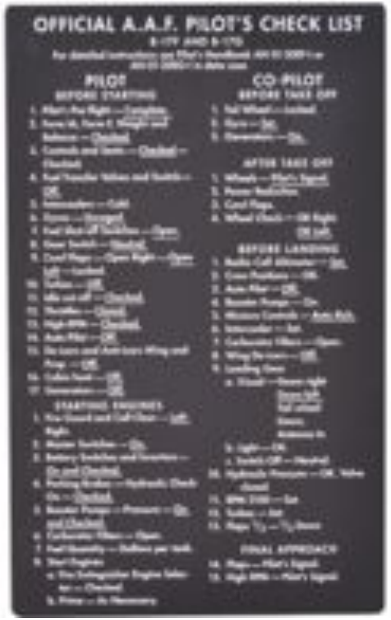
Code Review

- 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

Boeing Model 299 test on October 30, 1935.

- Plane crashed because of locked elevator control surface (opposite effect of MCAS)

Checklists help manage complex processes



OFFICIAL A.A.F. PILOT'S CHECK LIST

B-17F AND B-17G

For detailed instructions see Pilot's Handbook AN 01-20EF-1 or AN 01-20EG-1 in data case.

PILOT

BEFORE STARTING

1. Pilot's Pre-flight—Complete.
2. Form 1A, Form F, Weight and Balance—Checked.
3. Controls and Seats—Checked—Checked.
4. Fuel Transfer Valves and Switch—Off.
5. Intercoolers—Cold.
6. Gyros—Uncaged.
7. Fuel Shut-off Switches—Open.
8. Gear Switch—Neutral.
9. Cowl Flaps—Open Right—Open Left—Locked.
10. Turbos—Off.
11. Idle cut-off—Checked.
12. Throttles—Closed.
13. High RPM—Checked.
14. Auto Pilot—Off.
15. De-icers and Anti-icers Wing and Prop—Off.
16. Cabin Heat—Off.
17. Generators—Off.

STARTING ENGINES

1. Fire Guard and Call Clear—Left, Right.
2. Master Switches—On.
3. Battery Switches and Inverters—On and Checked.
4. Parking Brakes—Hydraulic Check On—Checked.
5. Booster Pumps—Pressure—On and Checked.
6. Carburetor Filters—Open.
7. Fuel Quantity—Gallons per tank.
8. Start Engines:
   a. Fire Extinguisher Engine Selector—Checked.
   b. Prime—As Necessary.

CO-PILOT

BEFORE TAKE-OFF

1. Tail Wheel—Locked.
2. Gyro—Set.
3. Generators—On.

AFTER TAKE-OFF

1. Wheels—Pilot's Signal.
2. Power Reduction.
3. Cowl Flaps.
4. Wheel Check—OK Right—OK Left.

BEFORE LANDING

1. Radio Call, Altimeter—Set.
2. Crew Positions—OK.
3. Auto Pilot—Off.
4. Booster Pumps—On.
5. Mixture Controls—Auto Rich.
6. Intercooler—Set.
7. Carburetor Filters—Open.
8. Wing De-icers—Off.
9. Landing Gear
   a. Visual—Down right, Down left, Tail wheel Down, Antenna in.
   b. Light—OK.
   c. Switch Off—Neutral.
10. Hydraulic Pressure—OK. Valve closed.
11. RPM 2100—Set.
12. Turbos—Set.
13. Flaps 1/3—1/3 Down.

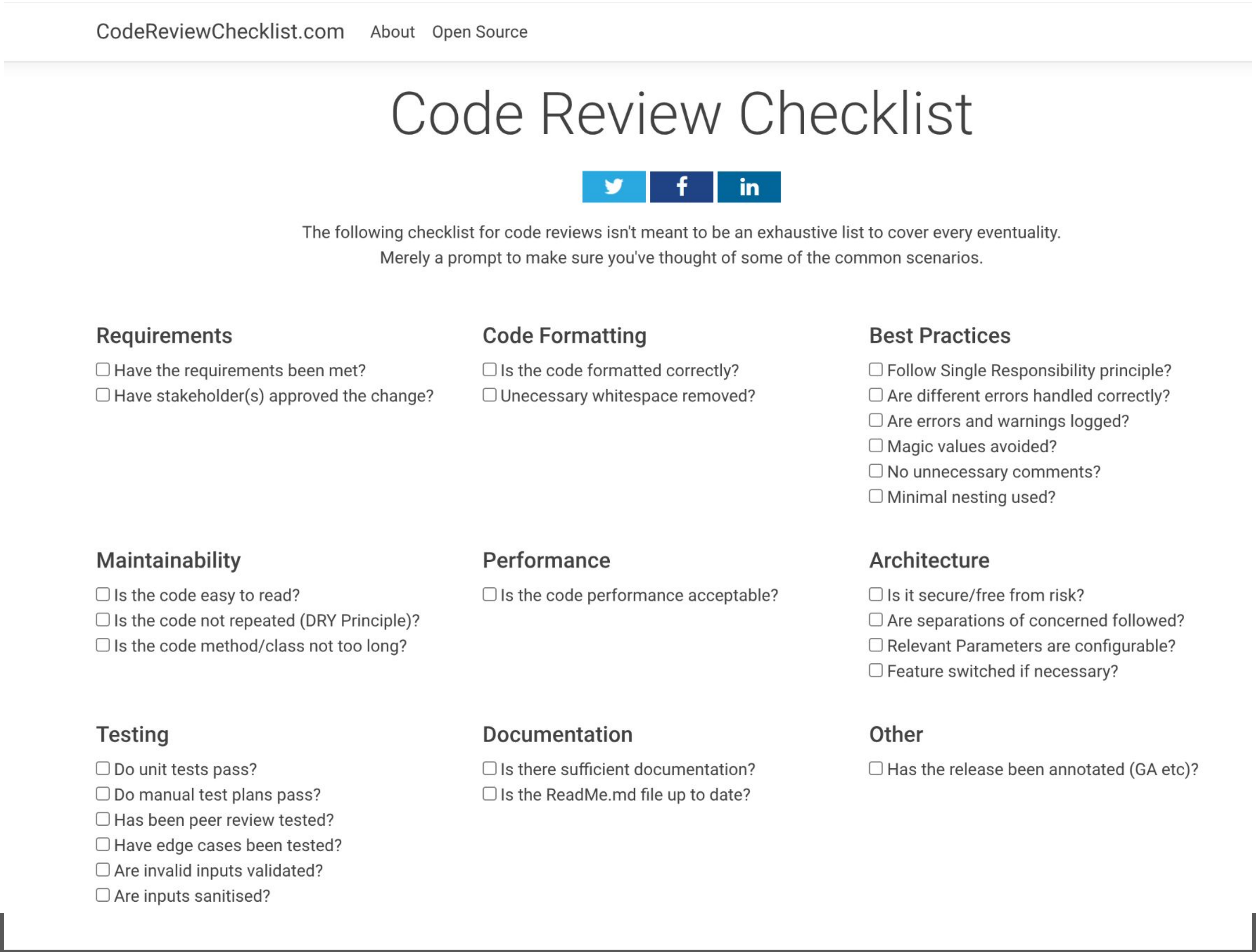
FINAL APPROACH

14. Flaps—Pilot's Signal.
15. High RPM—Pilot's Signal.

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

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”



CodeReviewChecklist.com About Open Source

Code Review Checklist

The following checklist for code reviews isn't meant to be an exhaustive list to cover every eventuality.
Merely a prompt to make sure you've thought of some of the common scenarios.

Requirements

□ Have the requirements been met?
□ Have stakeholder(s) approved the change?

Code Formatting

□ Is the code formatted correctly?
□ Unecessary whitespace removed?

Best Practices

□ Follow Single Responsibility principle?
□ Are different errors handled correctly?
□ Are errors and warnings logged?
□ Magic values avoided?
□ No unnecessary comments?
□ Minimal nesting used?

Maintainability

□ Is the code easy to read?
□ Is the code not repeated (DRY Principle)?
□ Is the code method/class not too long?

Performance

□ Is the code performance acceptable?

Architecture

□ Is it secure/free from risk?
□ Are separations of concerned followed?
□ Relevant Parameters are configurable?
□ Feature switched if necessary?

Testing

□ Do unit tests pass?
□ Do manual test plans pass?
□ Has been peer review tested?
□ Have edge cases been tested?
□ Are invalid inputs validated?
□ Are inputs sanitised?

Documentation

□ Is there sufficient documentation?
□ Is the ReadMe.md file up to date?

Other

□ Has the release been annotated (GA etc)?

Don’t forget that coders are people with feelings

- A coder’s self-worth is in their artifacts
- Continuous Integration 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