

ThoughtWorks®

AGILE FUNDAMENTALS

ThoughtWorks®

YOUR TRAINERS

ThoughtWorks®

KEVIN YEUNG

ThoughtWorks®

VIVEK PRAHLAD

SHORT INTRODUCTIONS

- Who are you?
- What do you do?
- What is your main learning objective?

SHORT INTRODUCTIONS

- Who are you?
- What do you do?
- What is your main learning objective?

EXERCISE

Agile experience



GROUND RULES

- Break times
- Restroom
- Start on time
- Phones off or on silent
- Laptops down
- No side conversations / one conversation at a time
- Criticize ideas, not people
- No hierarchy
- Respect

EXERCISE

Hopes and Fears



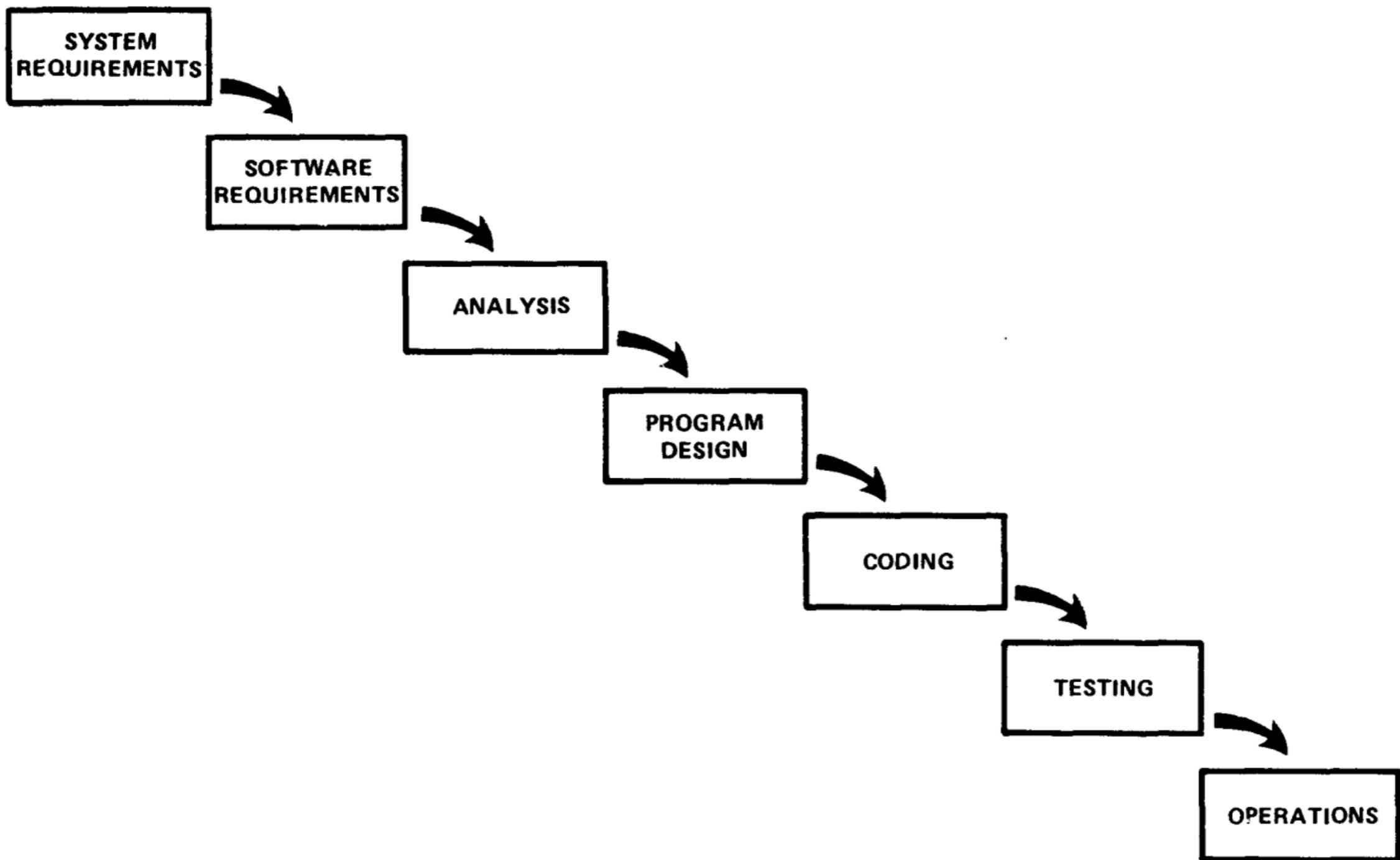
QUESTIONS?

ThoughtWorks®

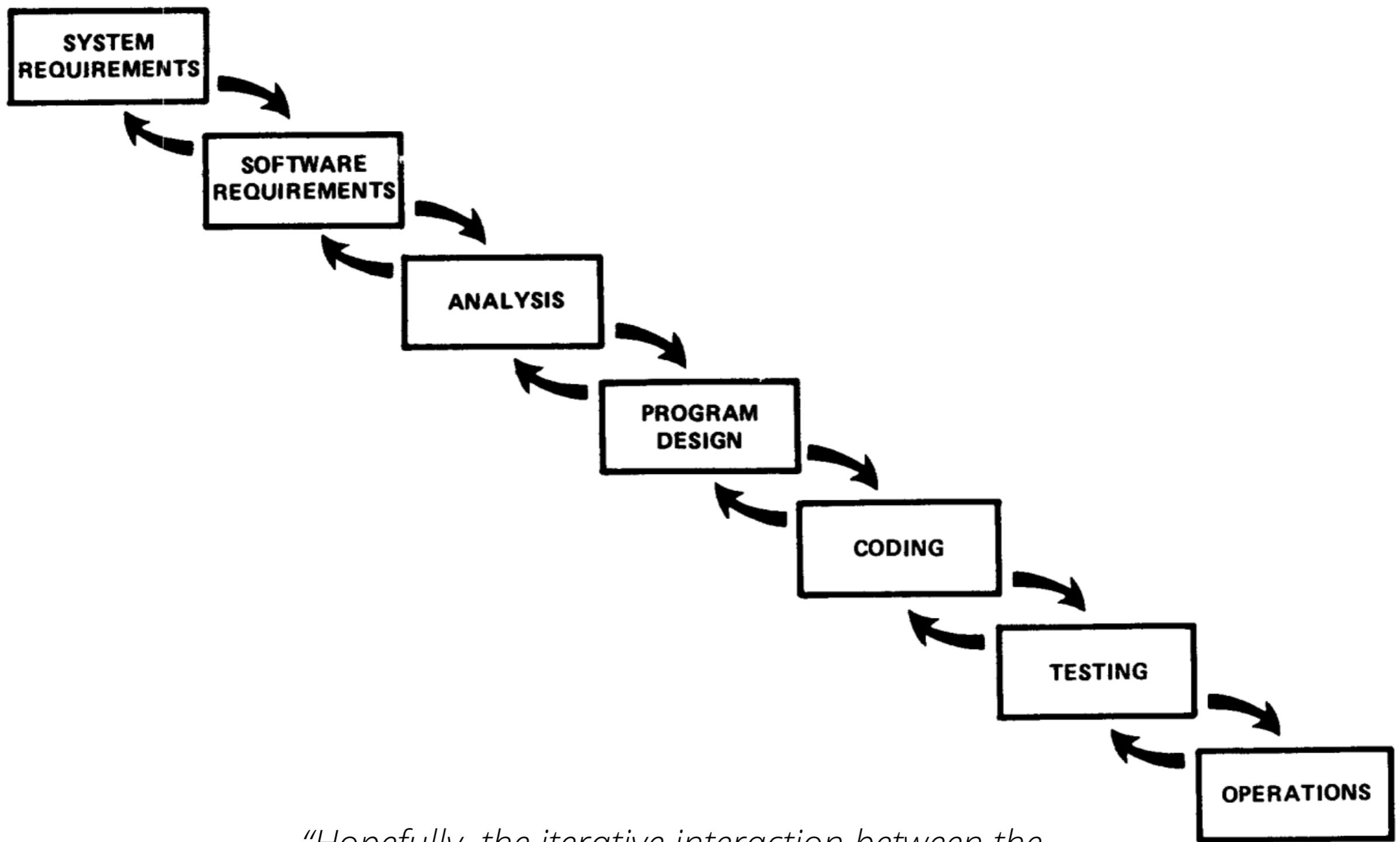
EXERCISE

Process doodle

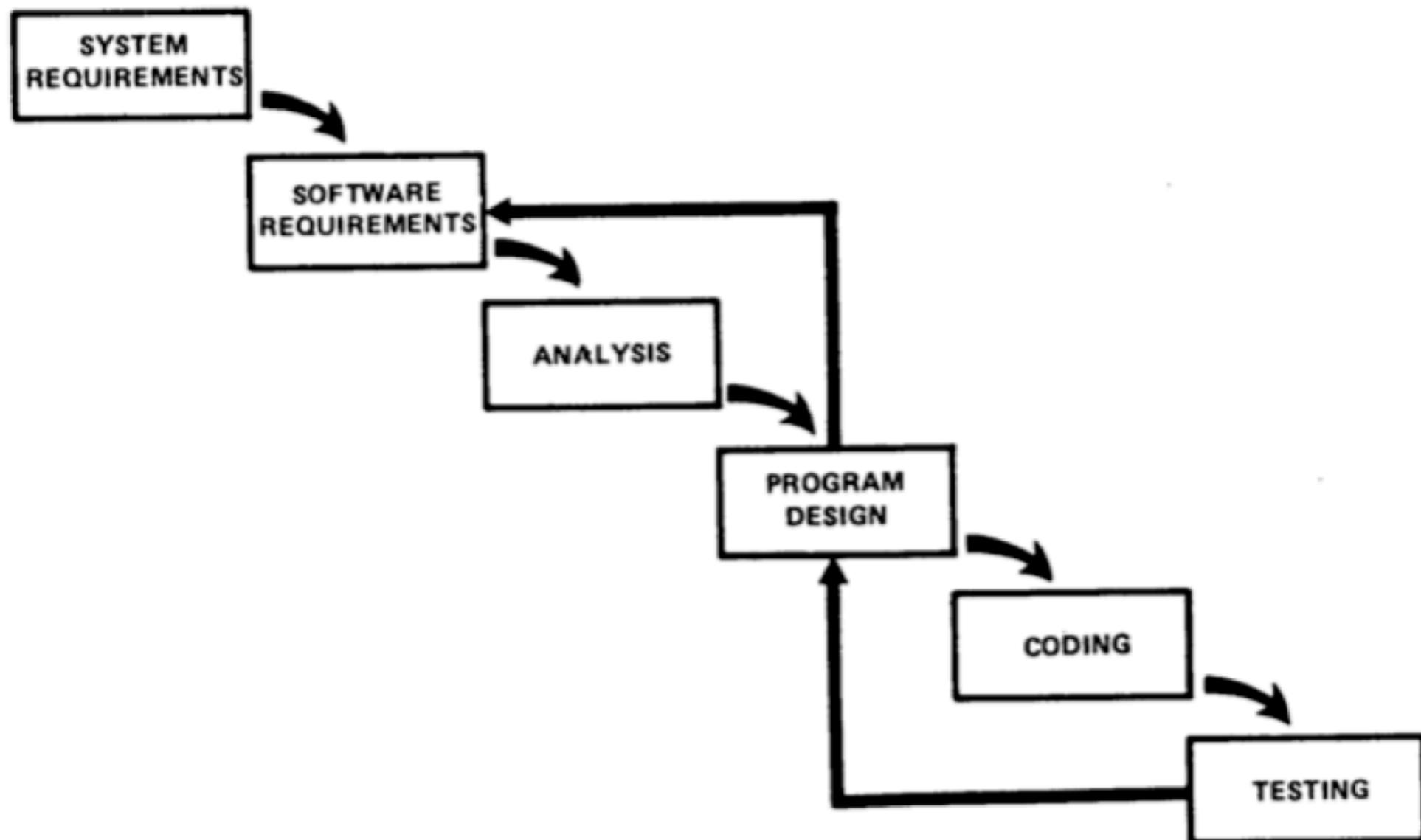




"I believe in this concept, but the implementation described above is risky and invites failure"

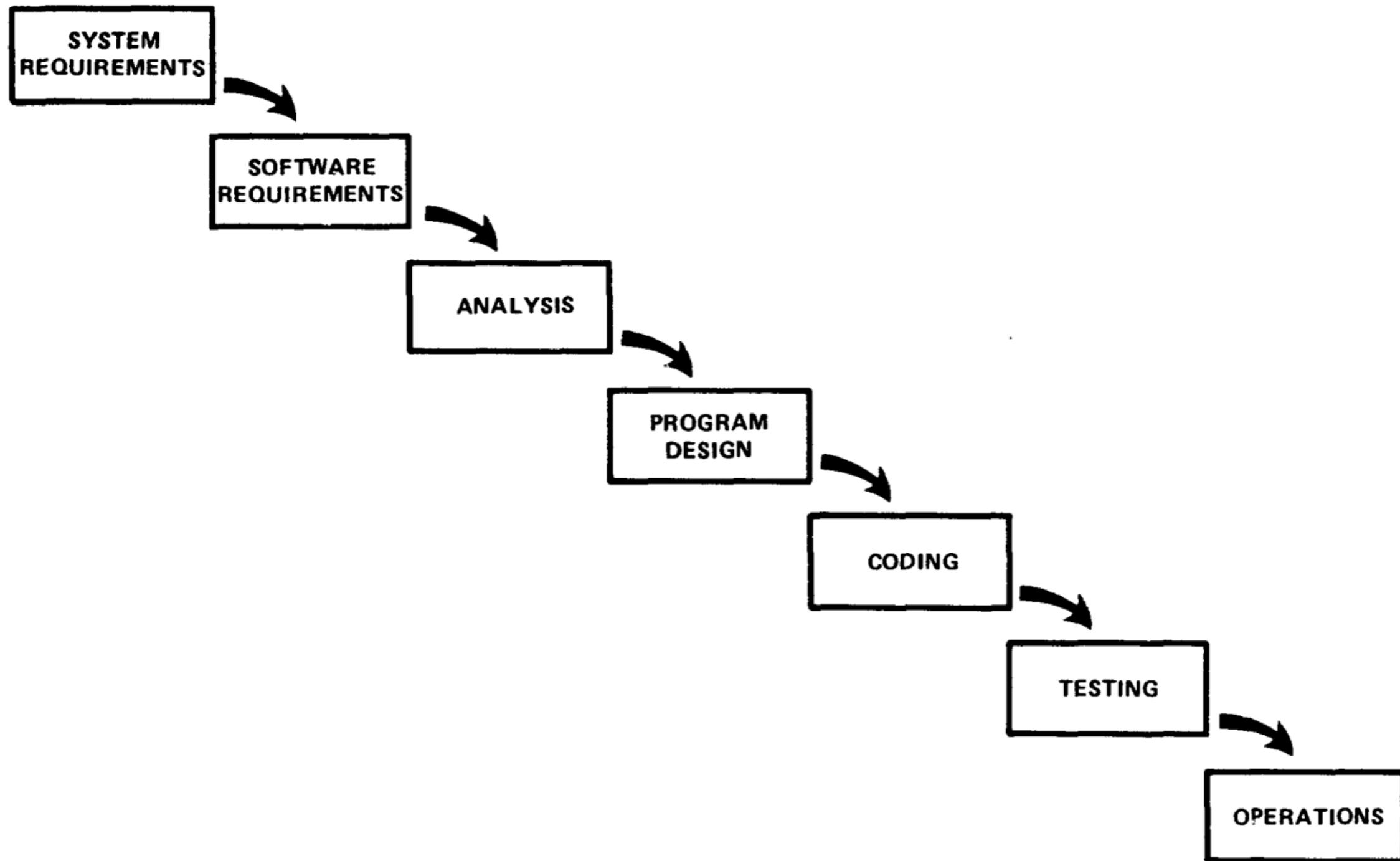


"Hopefully, the iterative interaction between the various phases is confined to successive steps."



I believe in this concept, but the implementation described above is risky and invites failure. The problem is illustrated in Figure 4

THE “WATERFALL” DIAGRAM



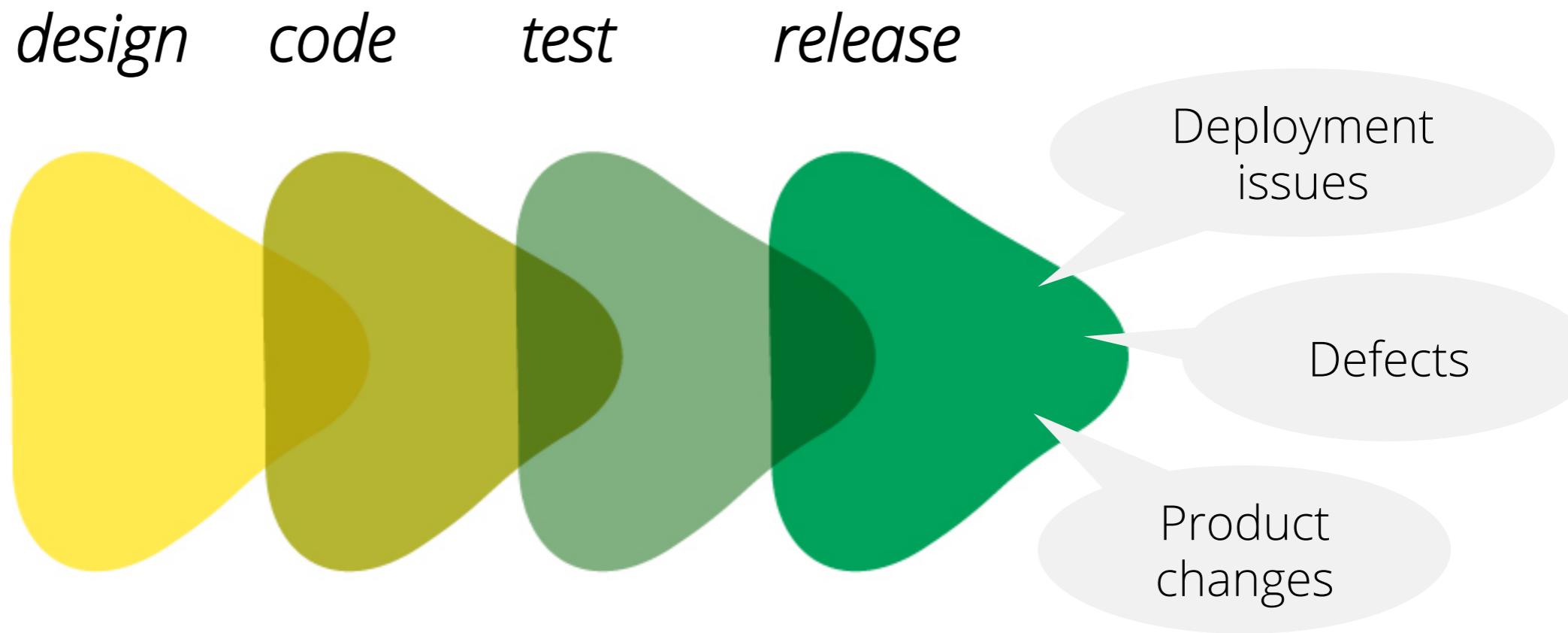
- [1] Royce, Winston. "Managing the Development of Large Software Systems", Proceedings of IEEE WESCON26 (August): 1–9. 1970.
[2] Bell, Thomas E., and T. A. Thayer. "Software requirements: Are they really a problem?", Proceedings of the 2nd international conference on Software engineering. IEEE Computer Society Press, 1976.

A WATERFALL RELEASE

design code test release

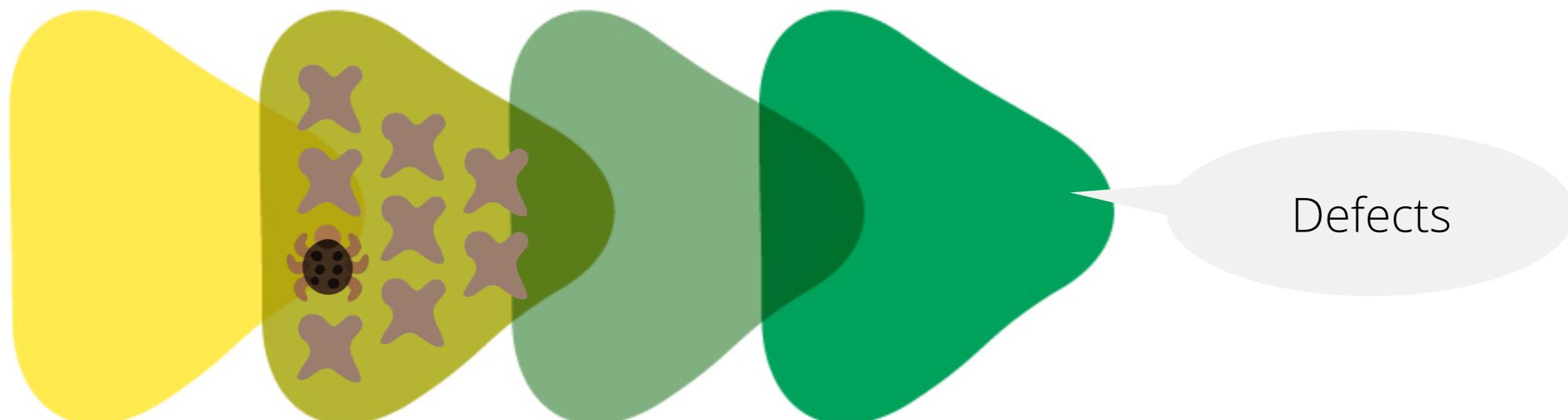


A WATERFALL RELEASE



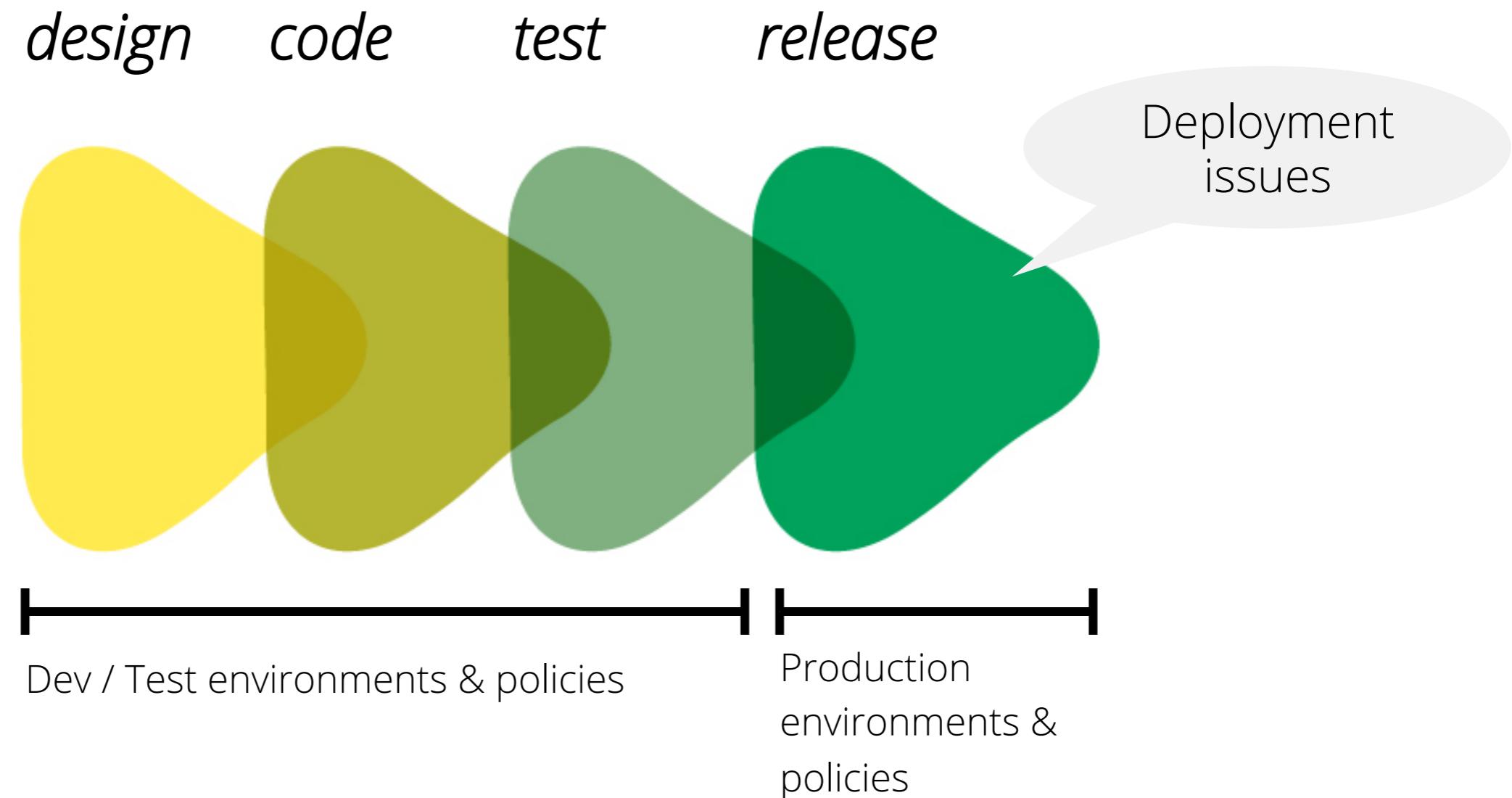
A WATERFALL RELEASE

design code test release

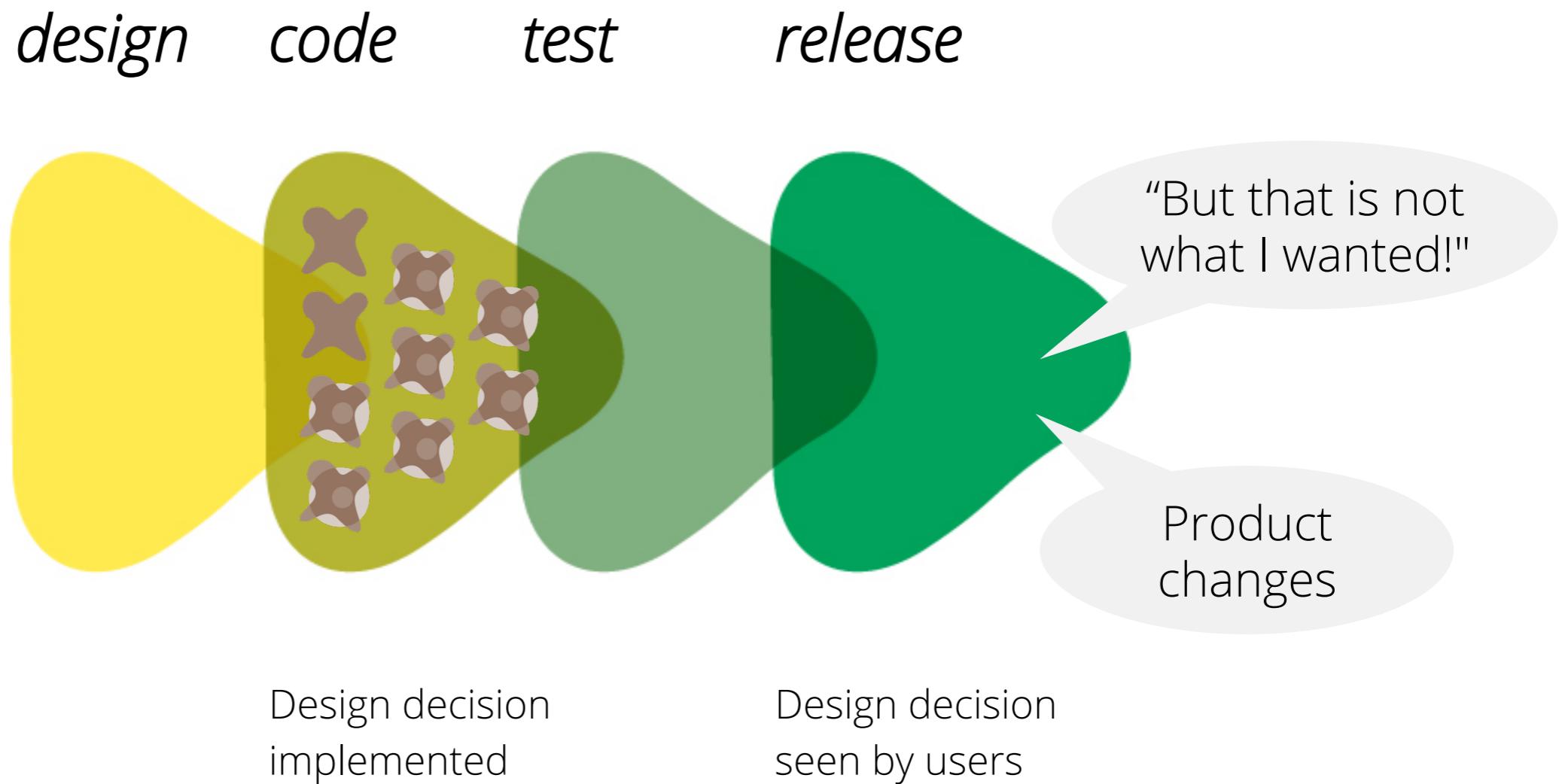


1. Defects introduced
2. Stuff built on top

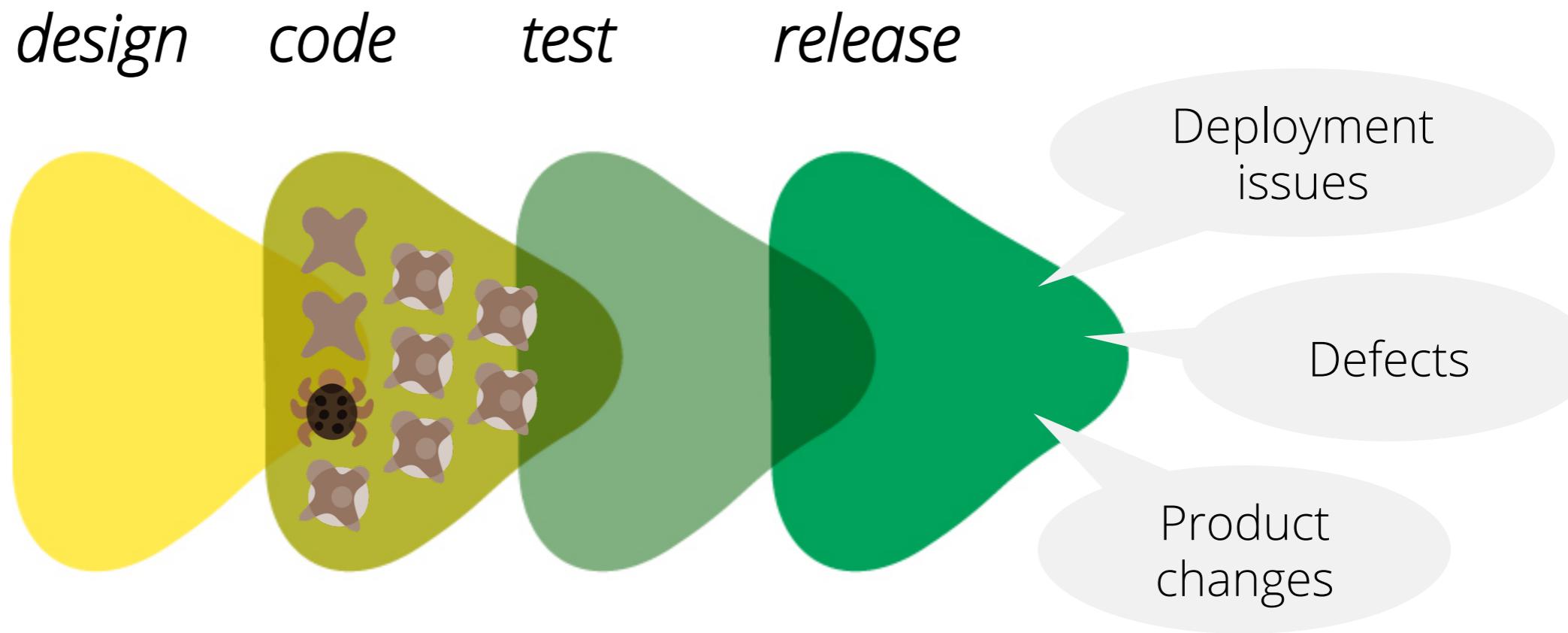
A WATERFALL RELEASE



A WATERFALL RELEASE



A WATERFALL RELEASE



ThoughtWorks®

AGILE SOFTWARE DEVELOPMENT

WHAT IS AGILE?

*A way of building software that is
all about planning for, and
assuming change*

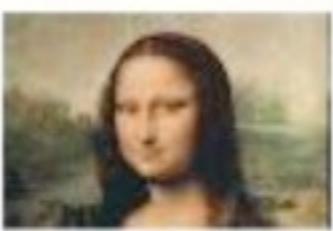
■ Incremental

Build software gradually

Show progress



1



2



3



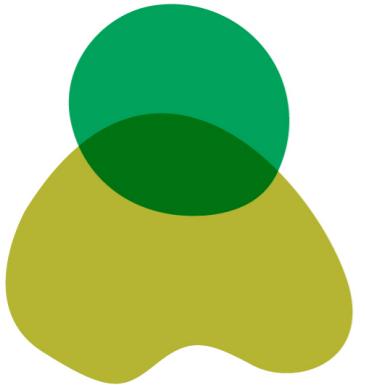
■ **Incremental**

Build software gradually

Show progress

■ **Iterative**

Multiple iterations and releases



woman in
pastoral
setting

1



2



3



■ **Incremental**

Build software gradually

Show progress

■ **Iterative**

Multiple iterations and releases

■ **Adaptive**

Incorporate feedback and changes to goals

Lessons learned

■ **Incremental**

Build software gradually

Show progress

■ **Iterative**

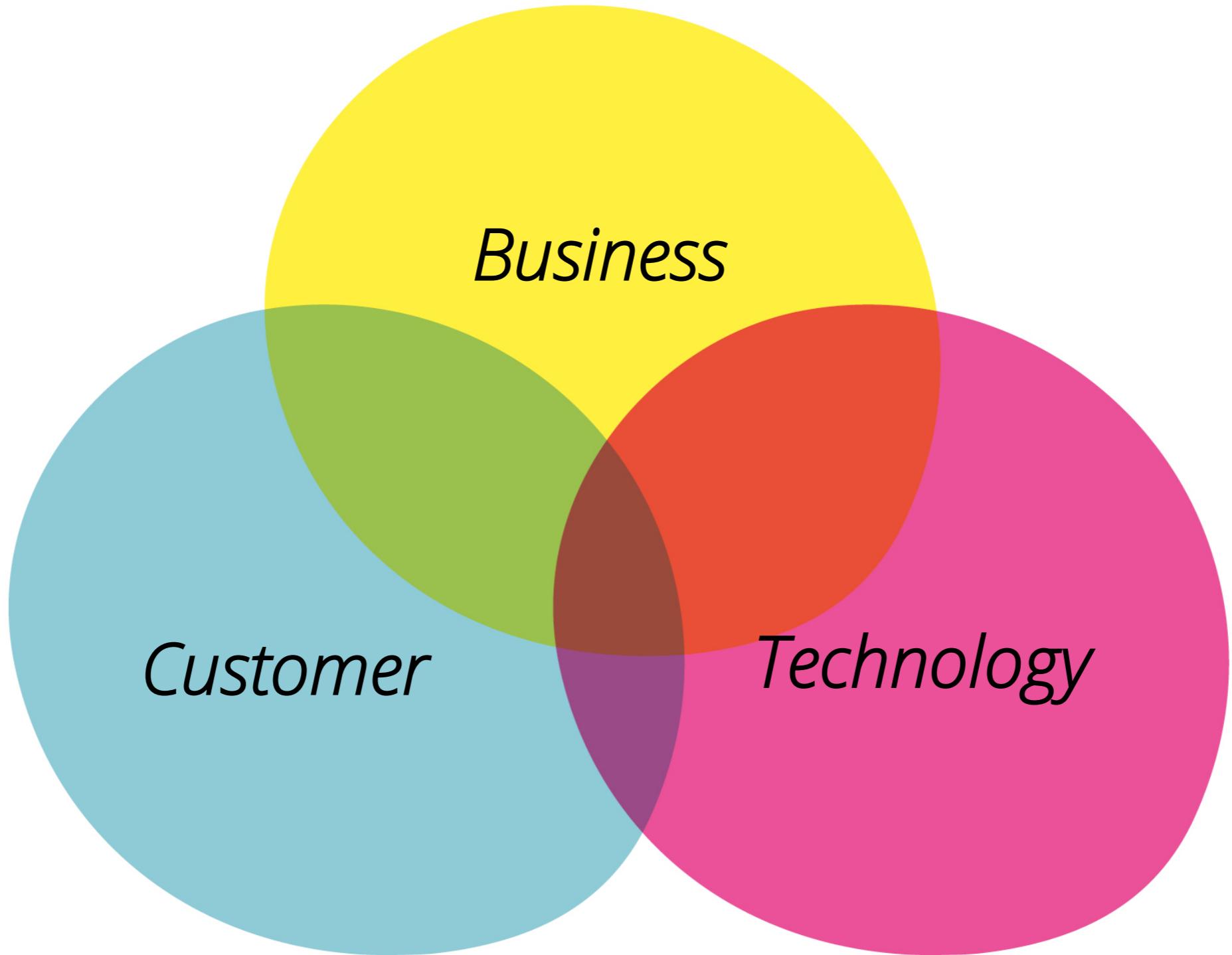
Multiple iterations and releases

■ **Adaptive**

Incorporate feedback and changes to goals

Lessons learned

■ **Collaborative**



EXERCISE

Agile Manifesto



MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT

Individuals and interactions *over* processes and tools

Working software *over* comprehensive documentation

Customer collaboration *over* contract negotiation

Responding to change *over* following a plan

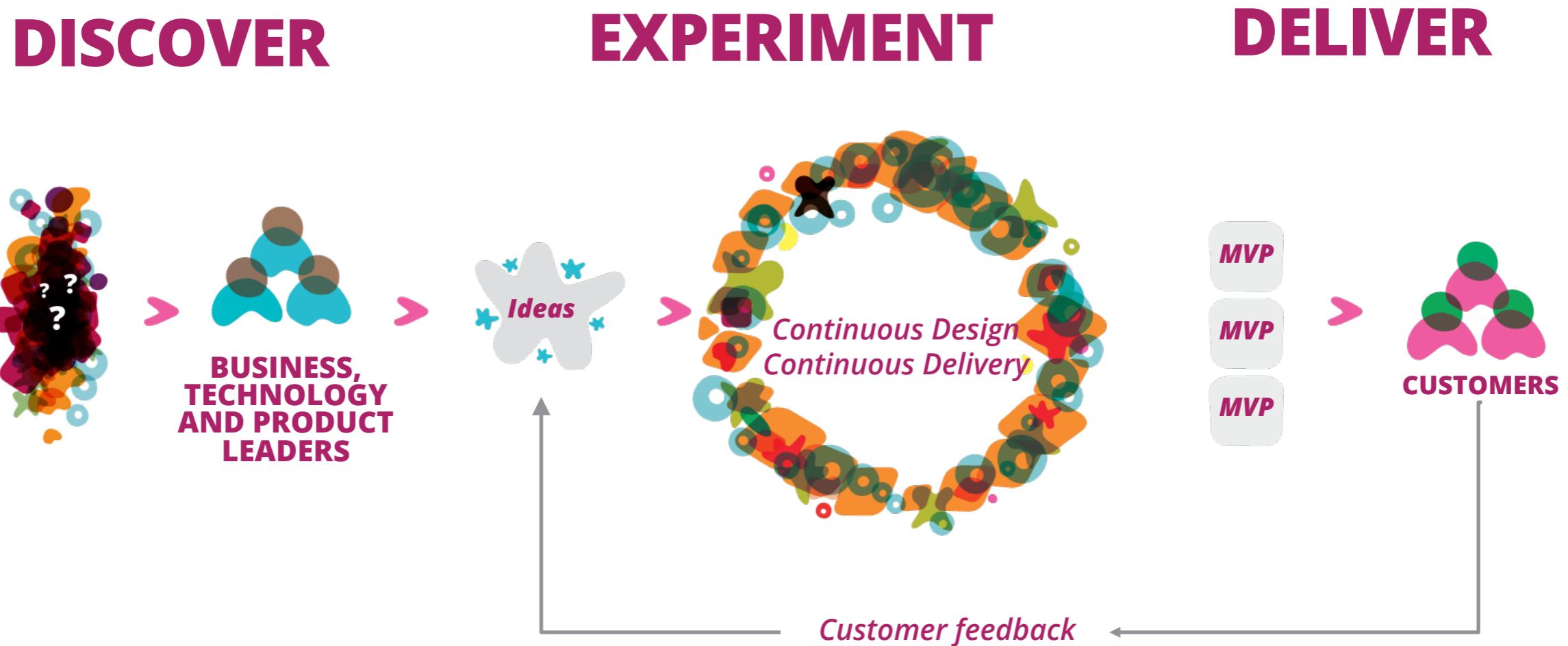
That is, while there is value in the items on the right, we value the items on the left more.

TWELVE PRINCIPLES OF AGILE SOFTWARE

1. Priority 1: Satisfy customer via early, continuous, valuable delivery
2. Welcome changing requirements, even late in development
3. Deliver working software frequently
4. Business people and developers must work together
5. Build projects around motivated individuals.
6. Face-to-face conversation is most effective way to communicate.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development.
9. Continuous attention to technical excellence and good design
10. Simplicity (art of maximizing the amount of work not done)
11. Emergent design and requirements from self-organizing teams
12. Regular reflection on how to become more effective

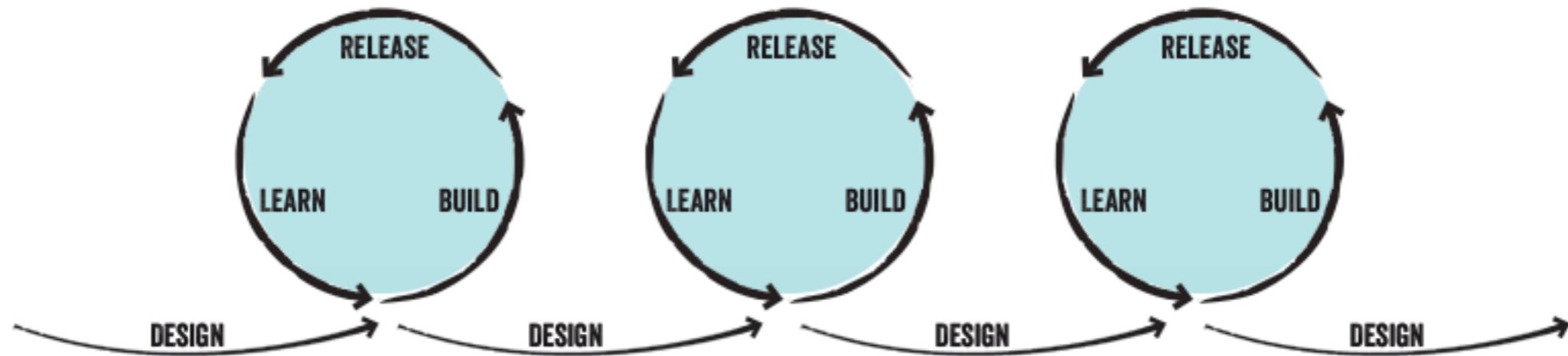
INTRODUCTION TO AGILE

OUR APPROACH

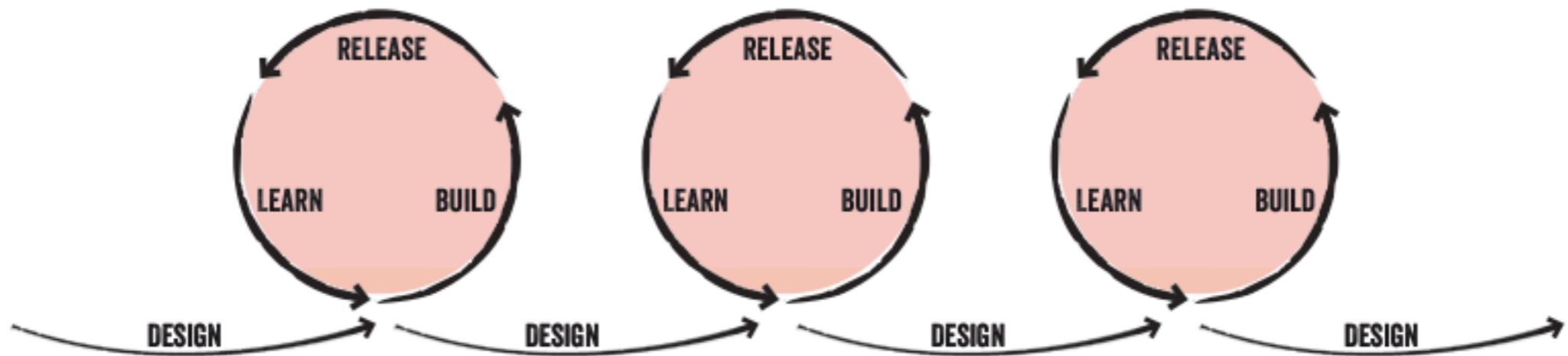


DESIGN THINKING AND AGILE

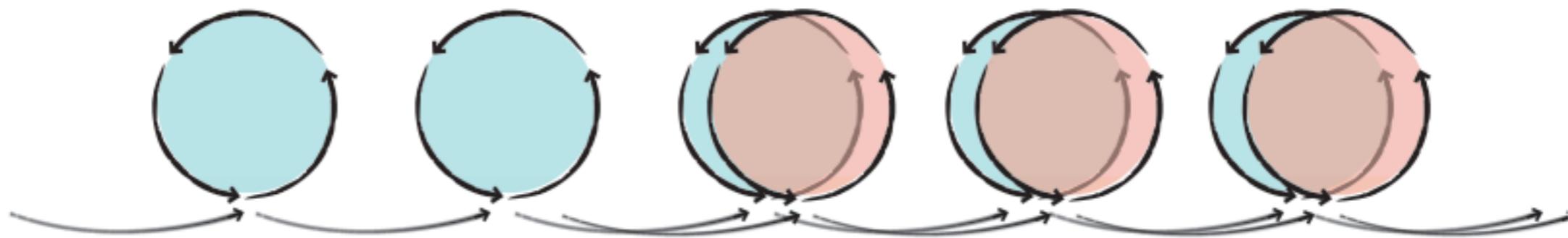
Design Thinking is the process of continually designing and learning from people (consumers, customers, patients, staff)



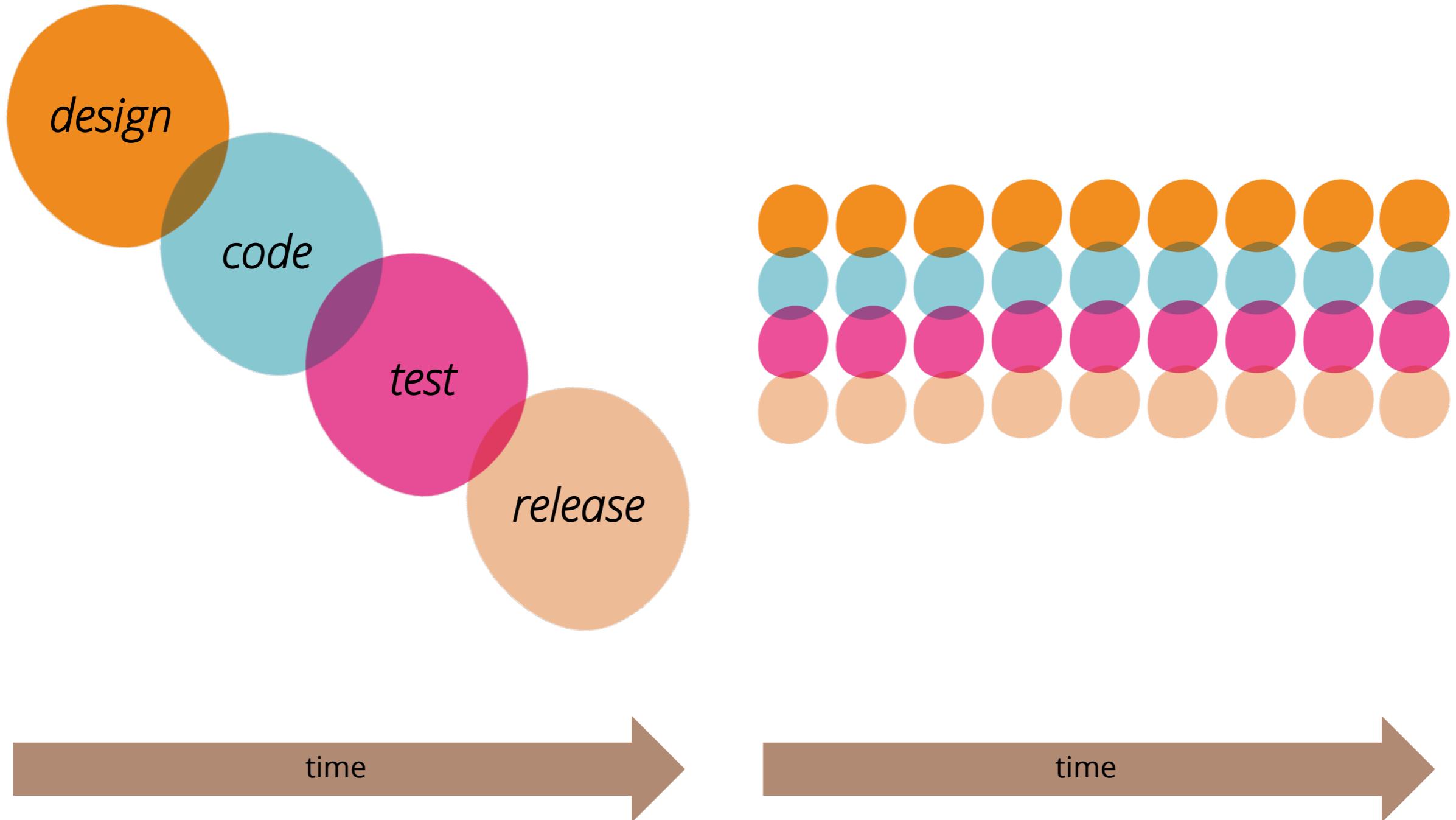
Agile Software Development is a process of rapid software design and build cycles which means constant releases to customers.



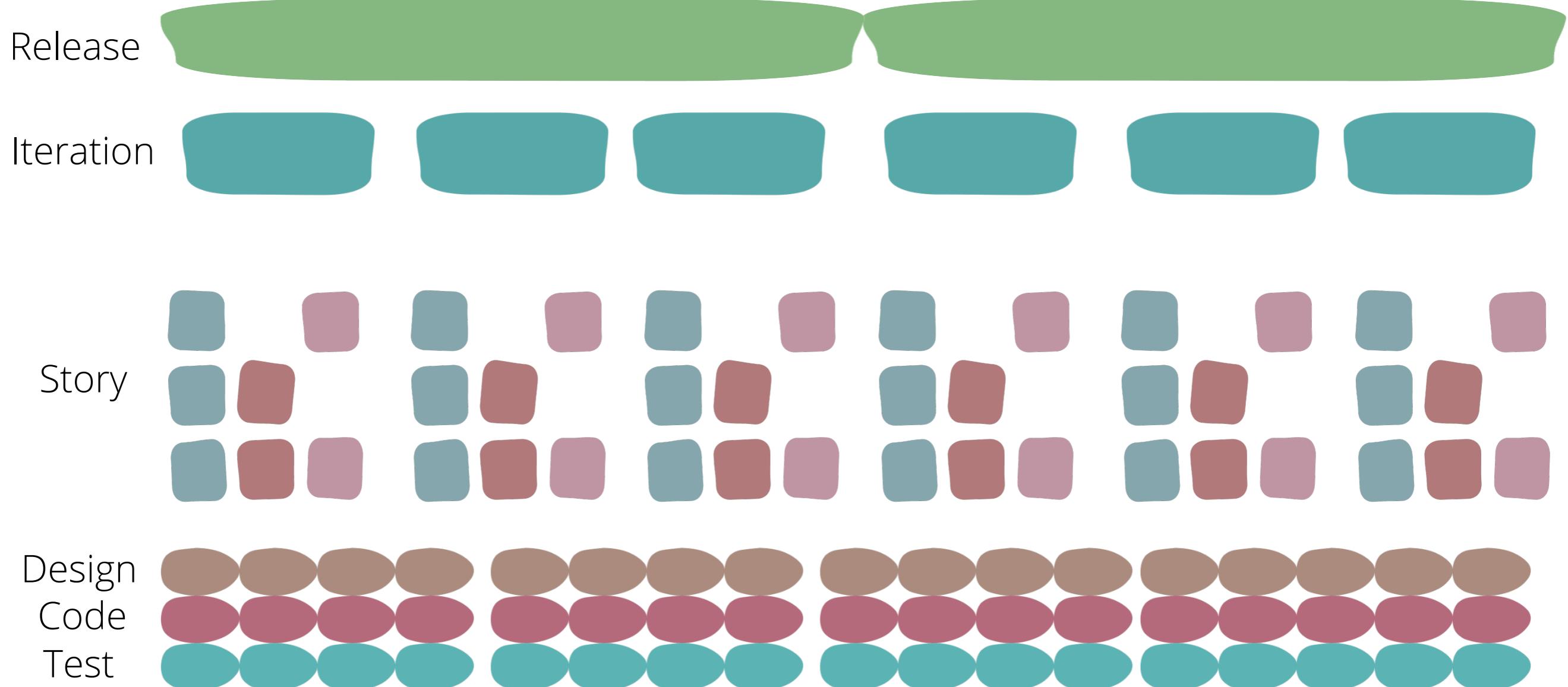
DESIGN THINKING AND AGILE



WATERFALL VS. AGILE APPROACH



FROM RELEASE TO STORY



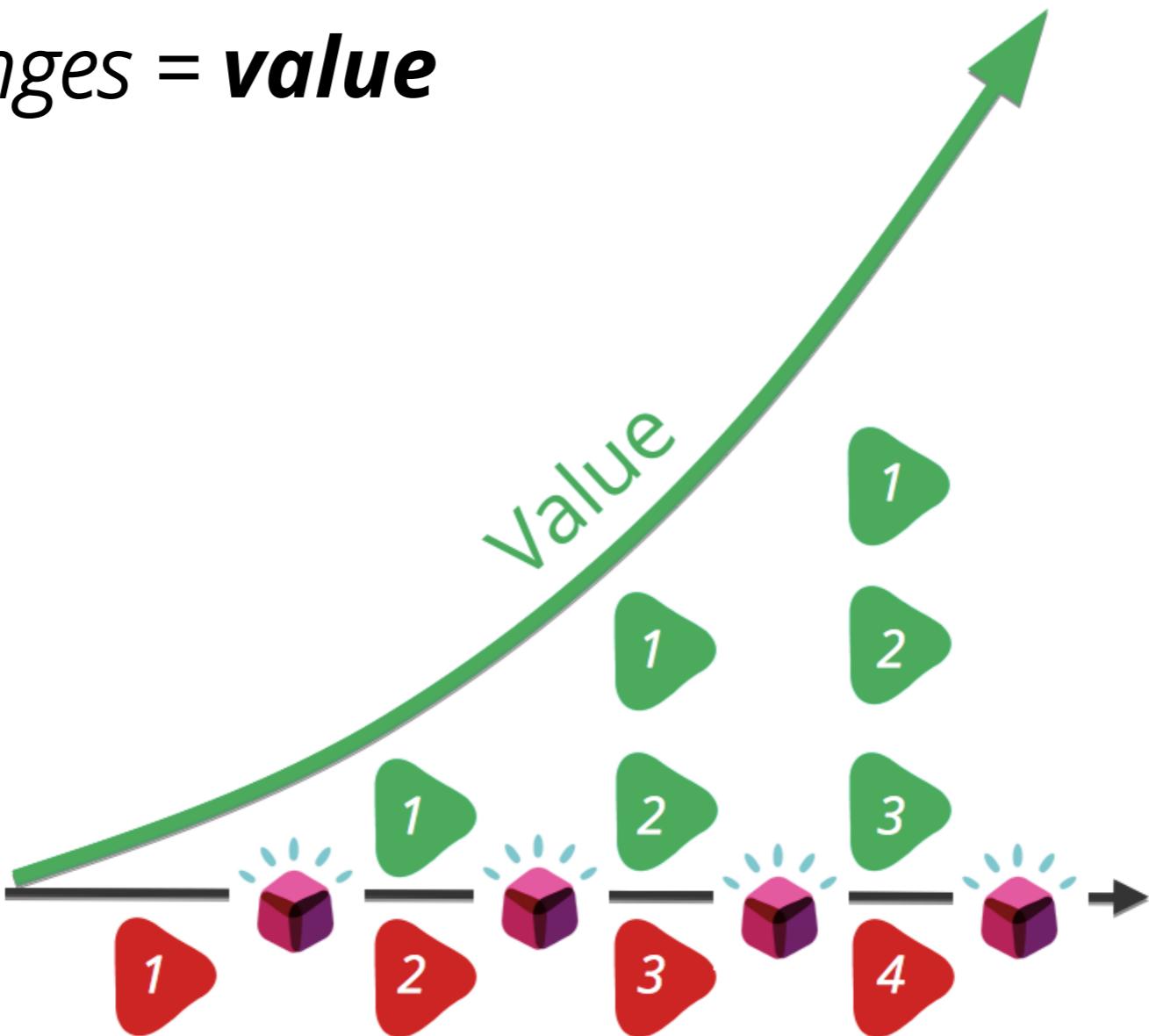
WATERFALL VS. AGILE APPROACH

unreleased changes = risk



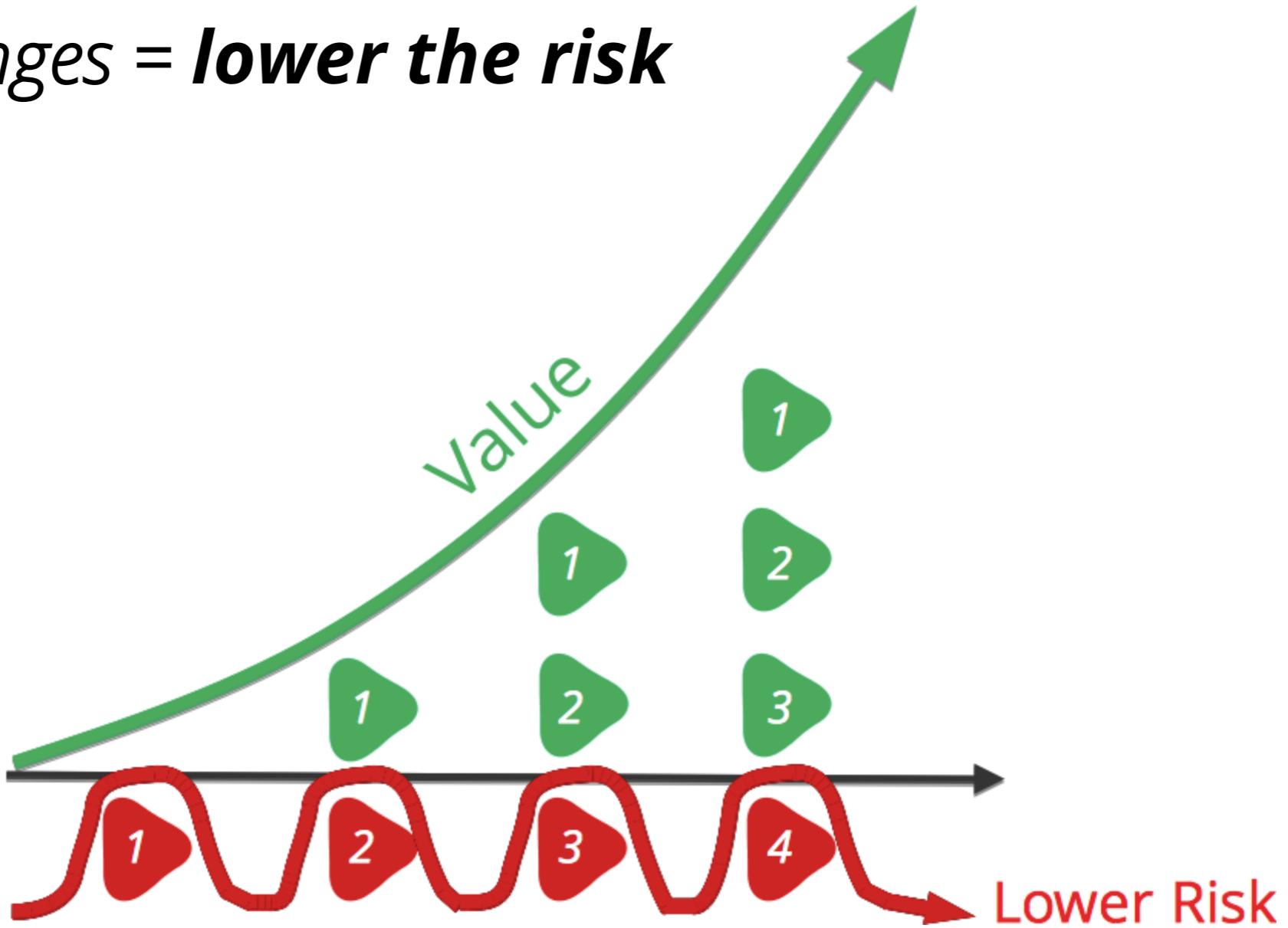
WATERFALL VS. AGILE APPROACH

released changes = value

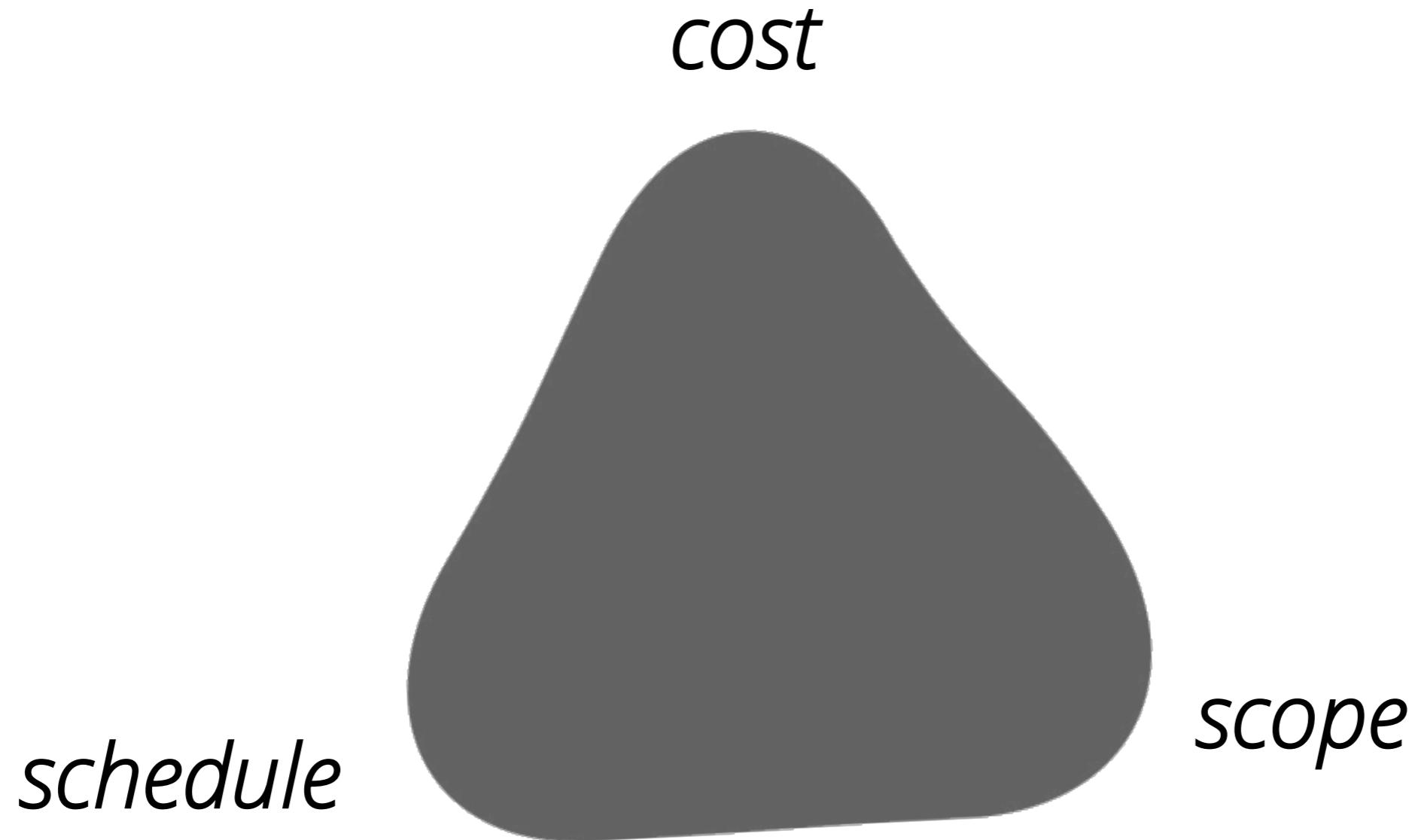


WATERFALL VS. AGILE APPROACH

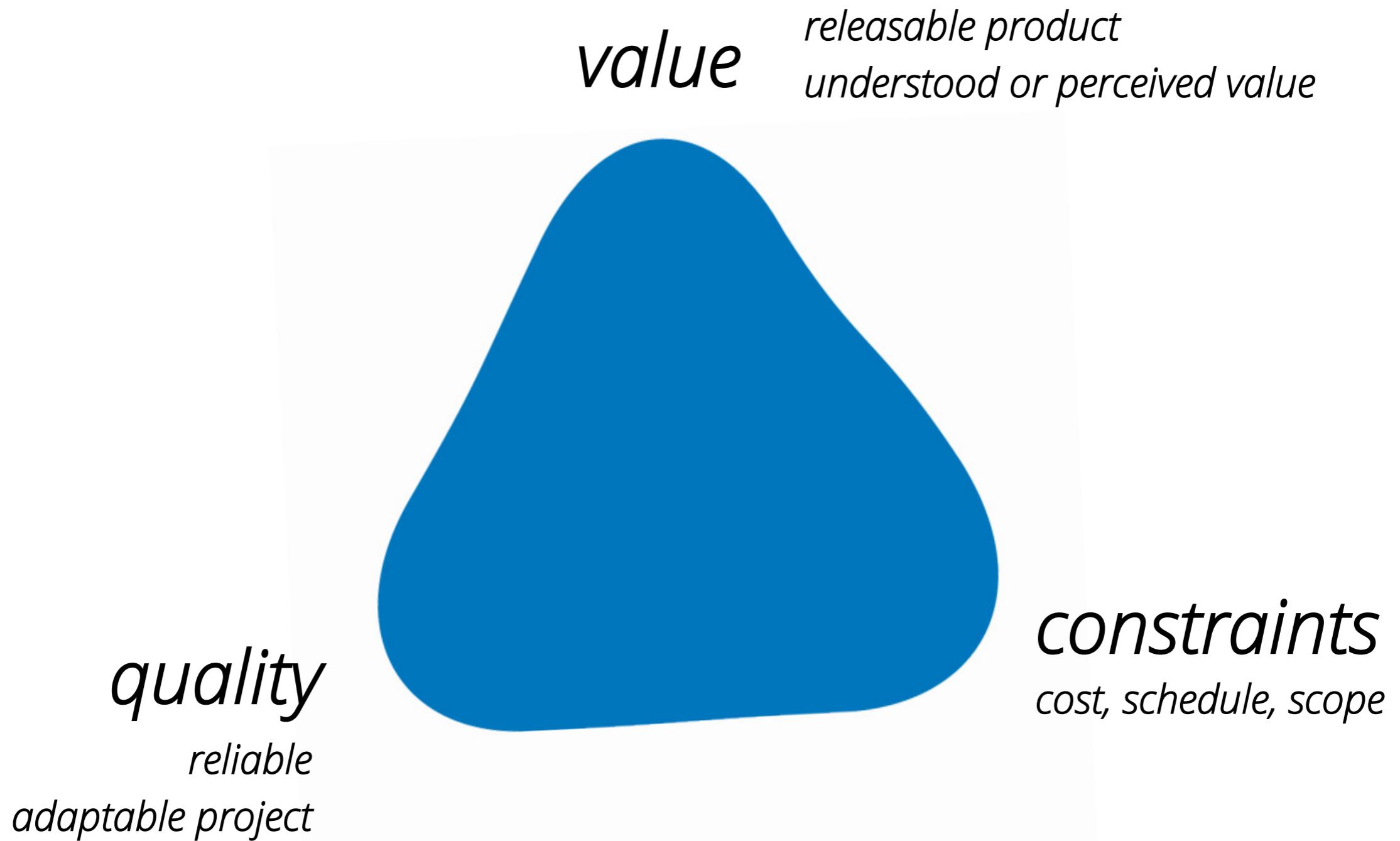
released changes = lower the risk



THE TRADITIONAL TRIANGLE



THE AGILE TRIANGLE



Both management and technical practices are important.

To be truly agile, the software itself needs to be agile.



WHAT AGILE IS *NOT*

IT IS NOT EITHER-OR

- Scrum
- XP (eXtreme Programming)
- Lean-kanban
- Hybrid (e.g., “Scrumban”)

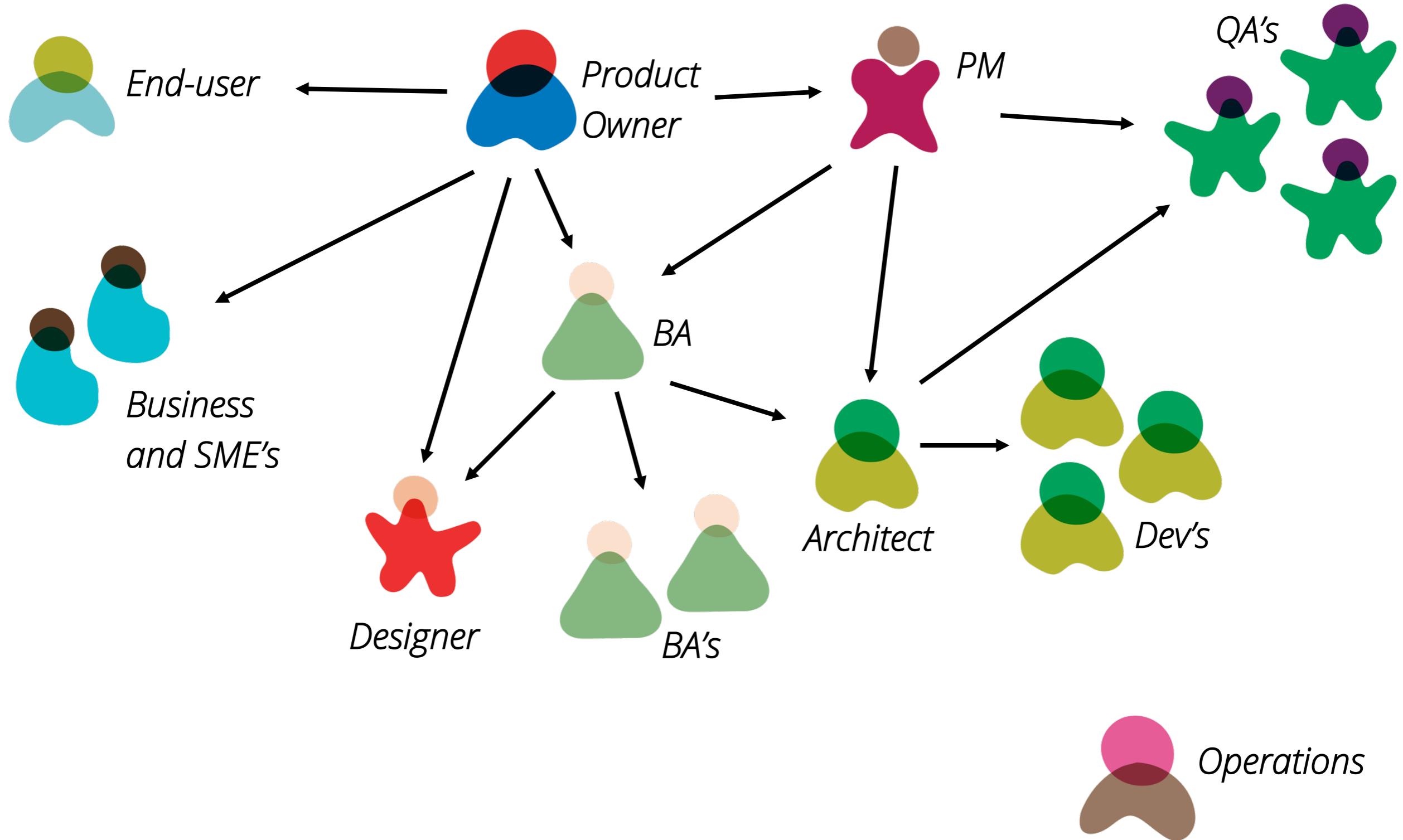
AGILE IS NOT...

- A “silver bullet”
- One size fits all
- Go through the motions and mechanics, and then “we are agile”
- Ad-hoc with no plan
- Anything can change anywhere, anytime



WHAT DO AGILE TEAMS LOOK LIKE?

TRADITIONAL SILOS





XD

PM

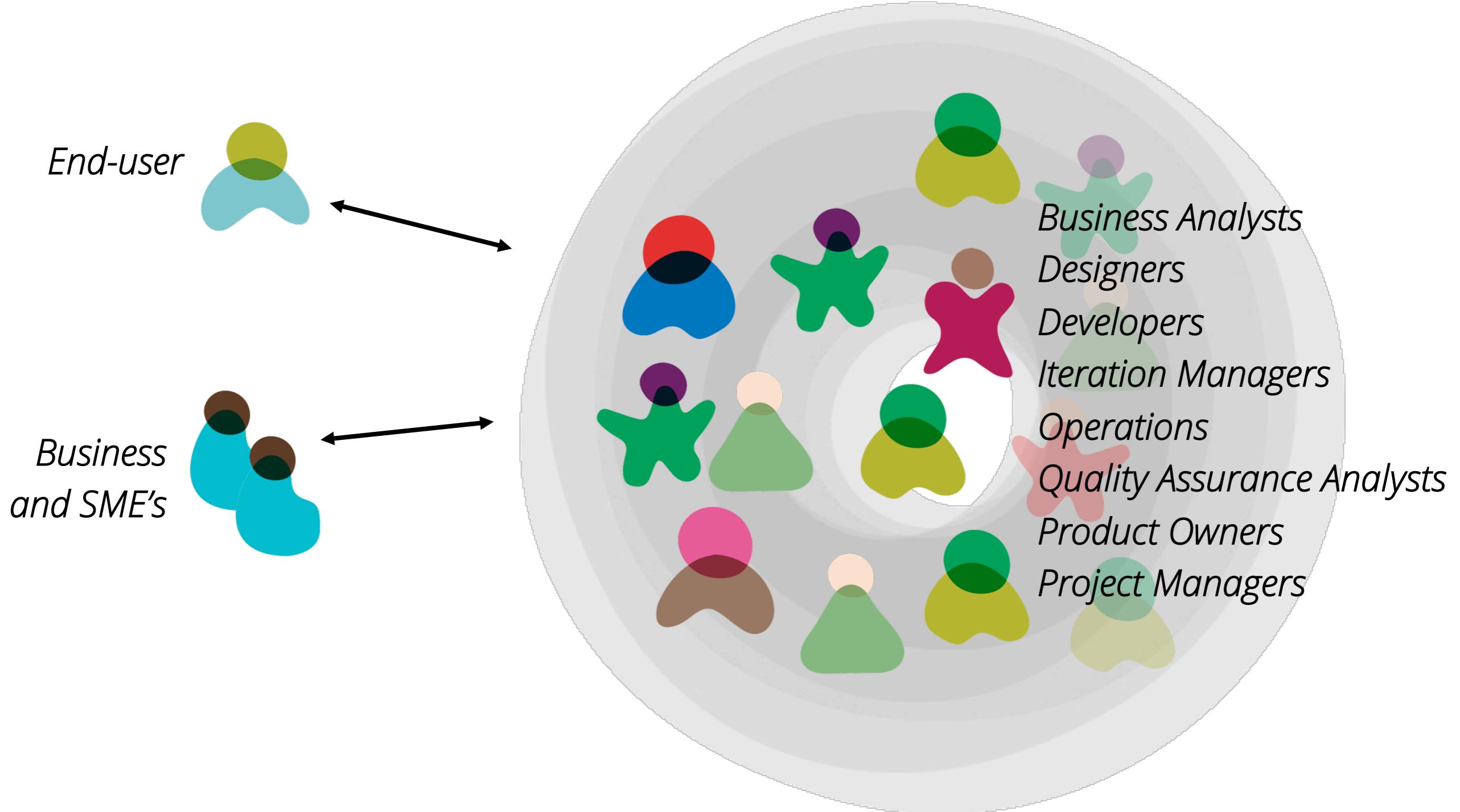
Dev
Ops

Dev

BA

Tester

EMPOWERED TEAMS



QUESTIONS?

ThoughtWorks®

USER STORIES

Light-weight requirements in Agile Projects



USER STORIES

- A tool for **iterative** development
- Represents a **unit** of work that should be developed
- Help **track** that piece of functionality's lifecycle
- Placeholder for a **conversation**

WHY SHOULD I USE THEM?

- It is a piece of customer-visible functionality written in common language
Universally understood
- Ensure you only build things for a reason
Prevents waste
- Simple and flexible
Minimum overhead
- Proven way of gathering requirements on agile projects
Effective teams

WHAT IS IN A USER STORY

- “As a ... I want ... So that ...”
- Acceptance Criteria
- Prototype
- User Interface design
- Other text/images/content to provide context

As a **<role>**

I want to **<business goal>**

So that **<value>**

INVEST IN STORIES

- I** Independent *No overlap - order is ok!*
- N** Negotiable *No contract. Details can change.*
- V** Valuable *Incremental benefit to something.*
- E** Estimable *Relative size to other stories.*
- S** Small *Shouldn't be bigger than an iteration.*
- T** Testable *Should be able to tell when it is done.*

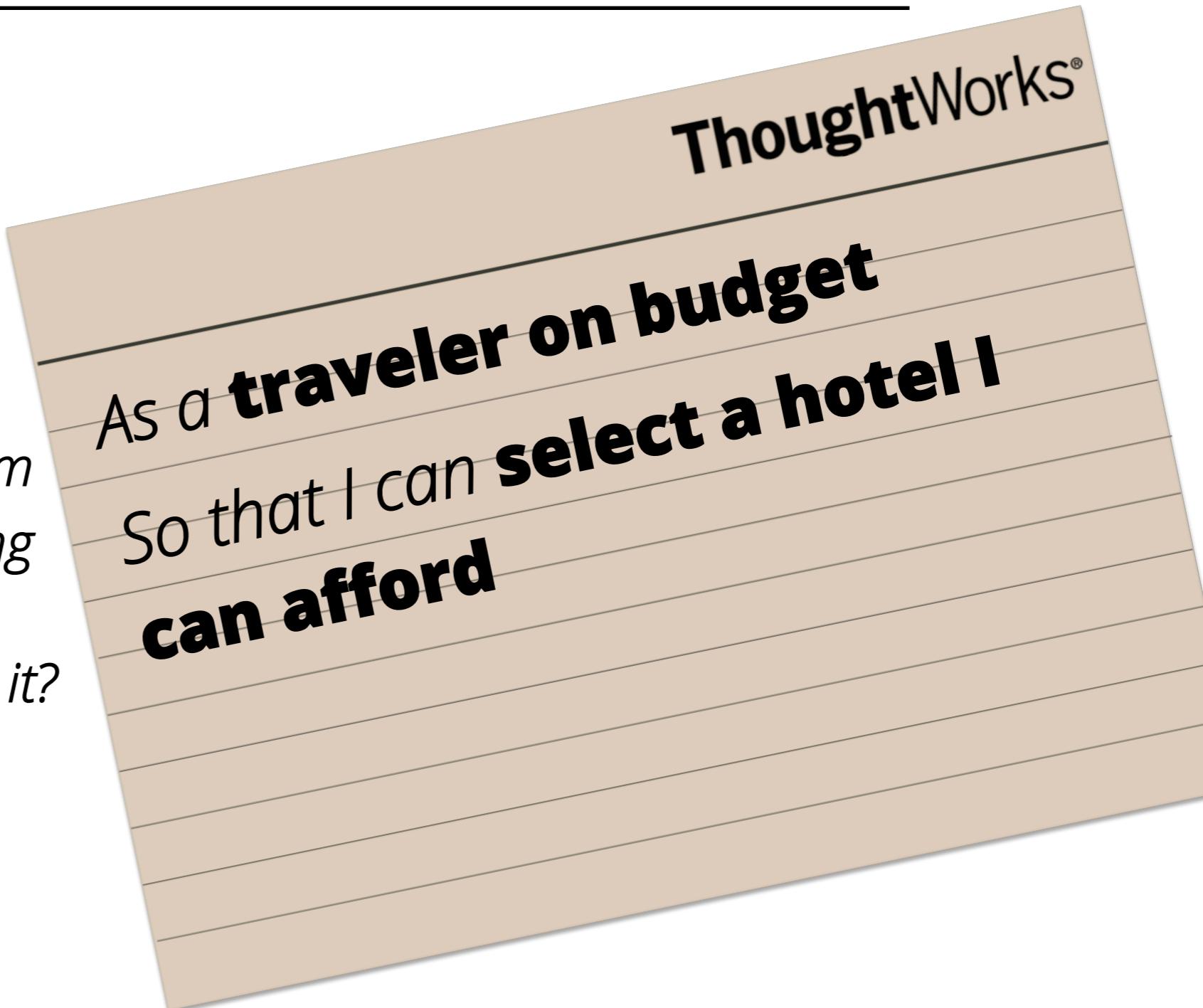
INVEST IN STORIES

V

Valuable

Someone is benefiting from what we are building

Why are we building it?



INVEST IN STORIES

V

Valuable

Someone is benefiting from what we are building

Why are we building it?



INVEST IN STORIES

V

Valuable

Someone is benefiting from what we are building

Why are we building it?



INVEST IN STORIES

T

Testable

Now we know what to do
and know when we will
be done



INVEST IN STORIES

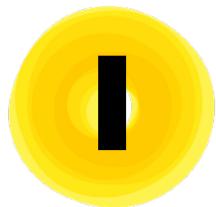


Small

*Just enough to get
feedback and avoid
waste*



INVEST IN STORIES



I Independent

Should be able to realize the value of each single story.

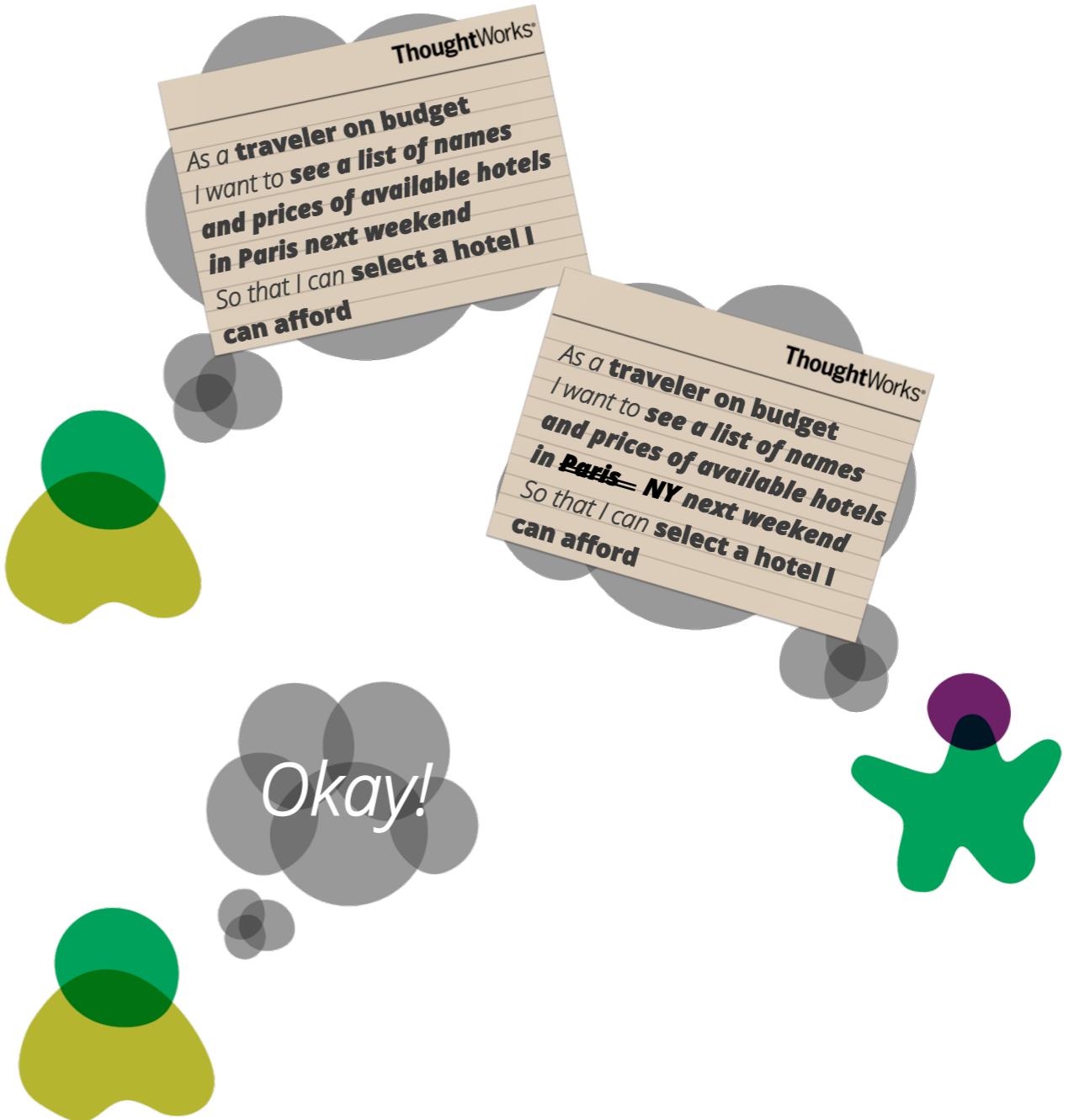
Distance from the airport can be in another story.



INVEST IN STORIES



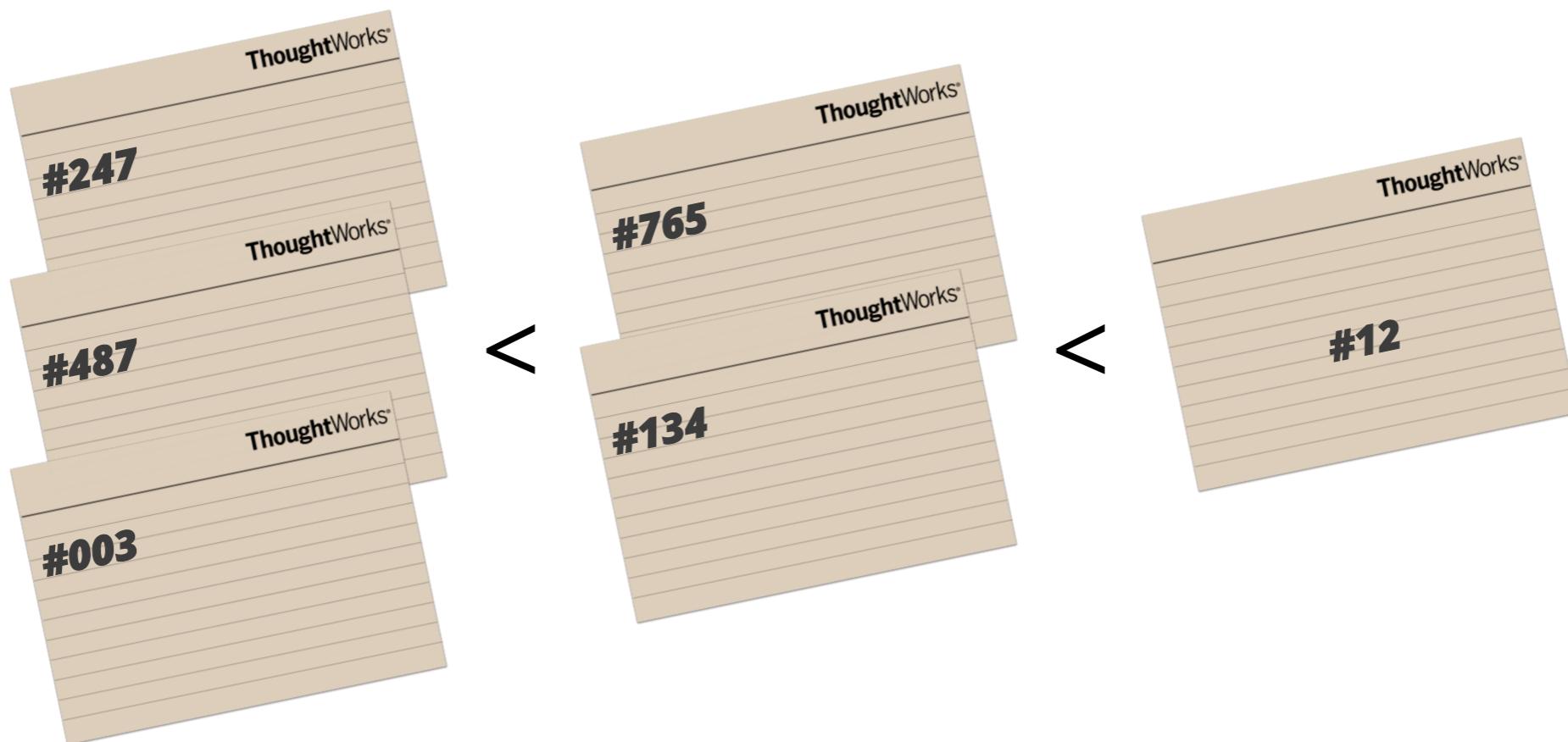
Negotiable



INVEST IN STORIES



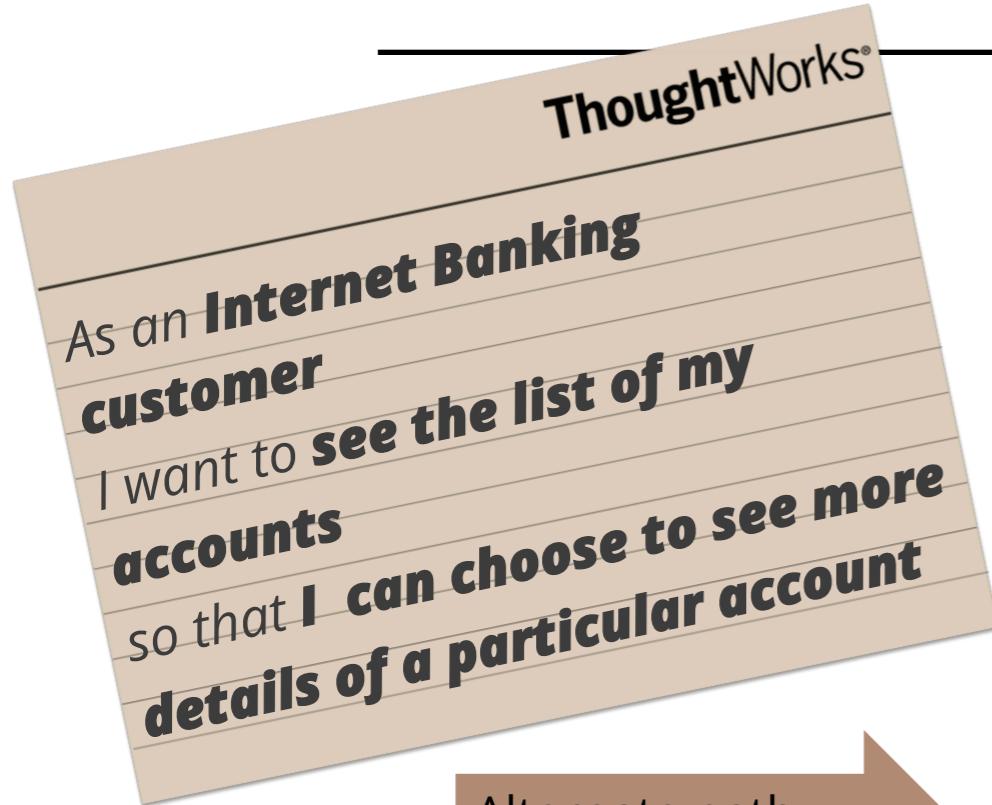
Estimable



ACCEPTANCE CRITERIA

- Tell you when a story is finished
- System is working when all AC's are accepted
- A story can have one or multiple AC's
- Acceptance criteria are automated
- Common format:
 - Give <context>
 - When <event>
 - Then <outcome>

ACCEPTANCE CRITERIA



Alternate path

Alternate path

Bad path

Given the customer has one transaction account and one credit account
When they have completed logging in

Then the screen should show the names and numbers of the two accounts sorted in account number order

Given the customer has just one transaction account

When they have completed logging in

Then the screen should show the name and number of the account

Given the customer has no accounts

When they have completed logging in

Then the screen should show a message stating that no accounts are available

Given the customer has more than 20 accounts

When they have completed logging in

Then the screen should show the first 20 accounts (in account number order) only

Given the customer has some accounts

And they have completed logging in

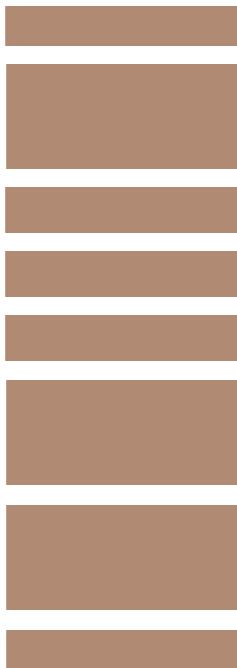
When the system cannot retrieve the account details

Then the screen should show an error message with associated code and details to contact for support

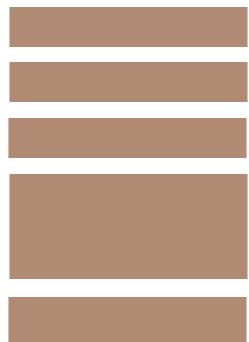
WHAT IS *NOT* IN A USER STORY

- Unit and Integration tests
- Documentation
- Specifications
- Architecture
- Code

STORY DETAILS AT THE RIGHT TIME



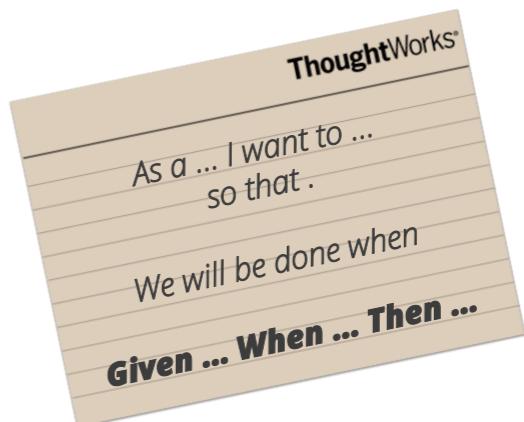
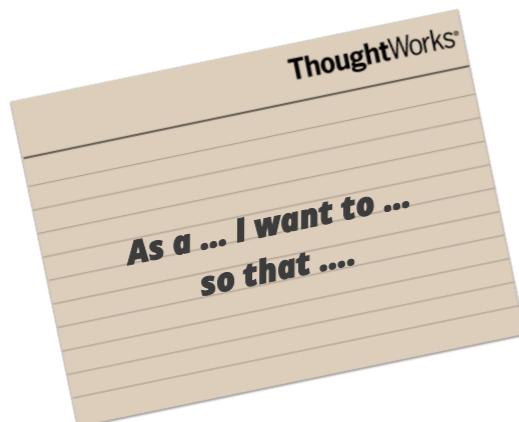
*Product
backlog*



*Release
backlog*



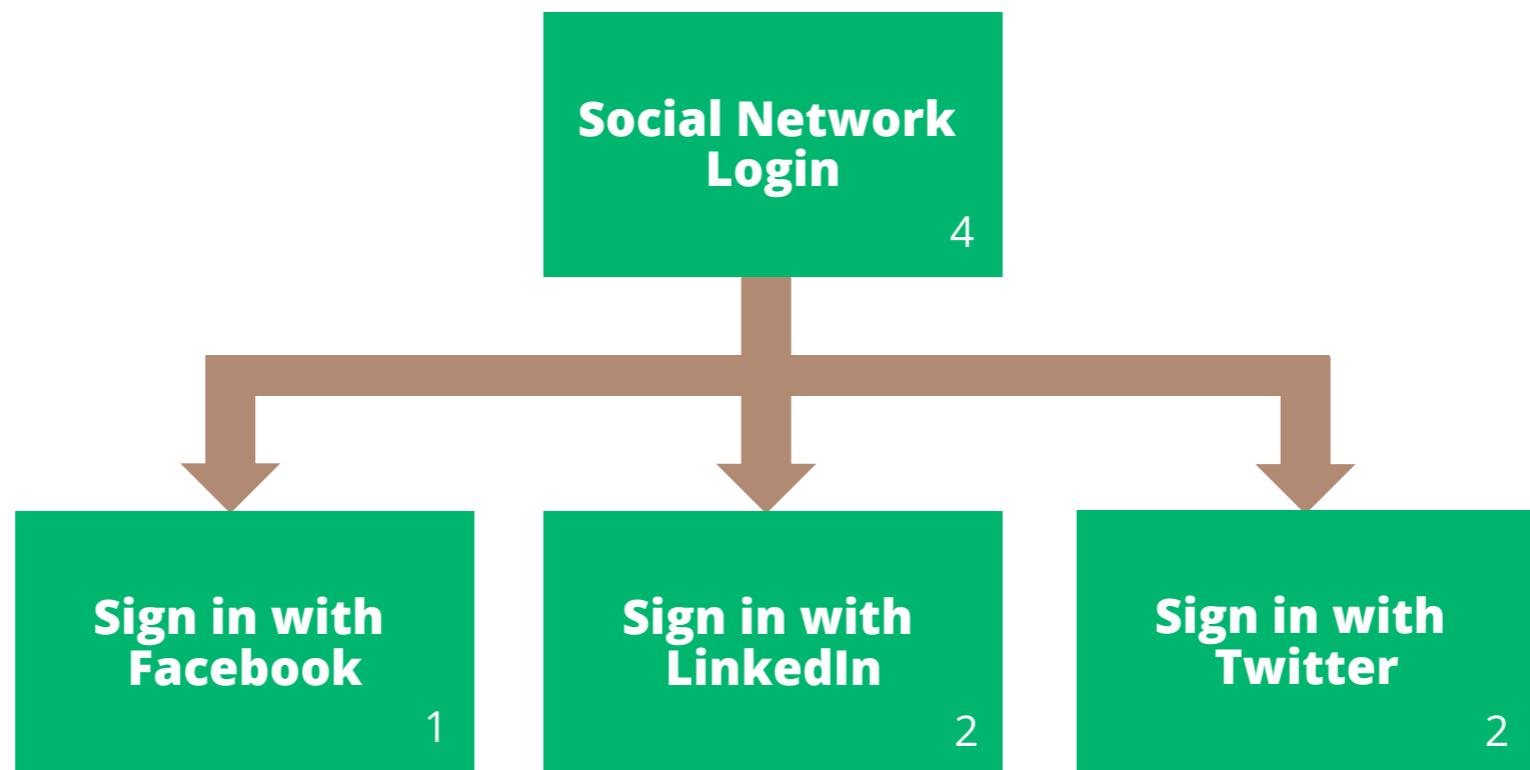
*Iteration
backlog*



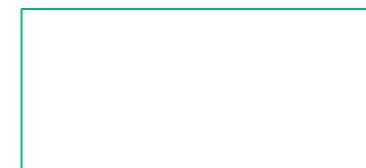
STORY BREAKDOWN

- Identify opportunities to split individual stories into one or more smaller independent stories, in order to:
 - Make stories an appropriate size for delivery
 - eg. *small enough that they are easily understandable, estimable and will give a good indication of progress*
 - Identify differences in priority
 - to ensure only the highest priority work is completed first*

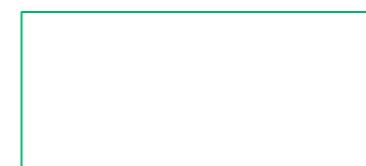
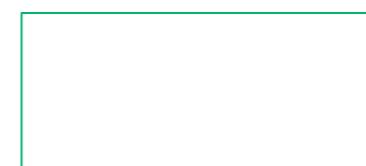
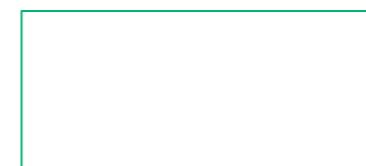
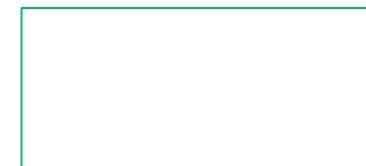
SPLITTING BY FEATURE



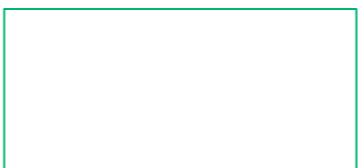
SPLITTING BY PRIORITY



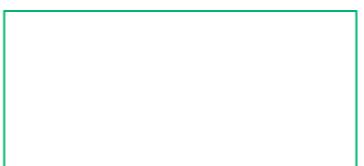
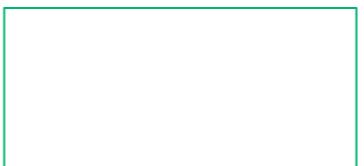
**Social
Network
Login**



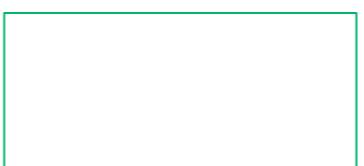
*The resulting stories
are independently
prioritized, revealing
a range of priorities*



**Sign in with
Facebook**



**Sign in with
Twitter**

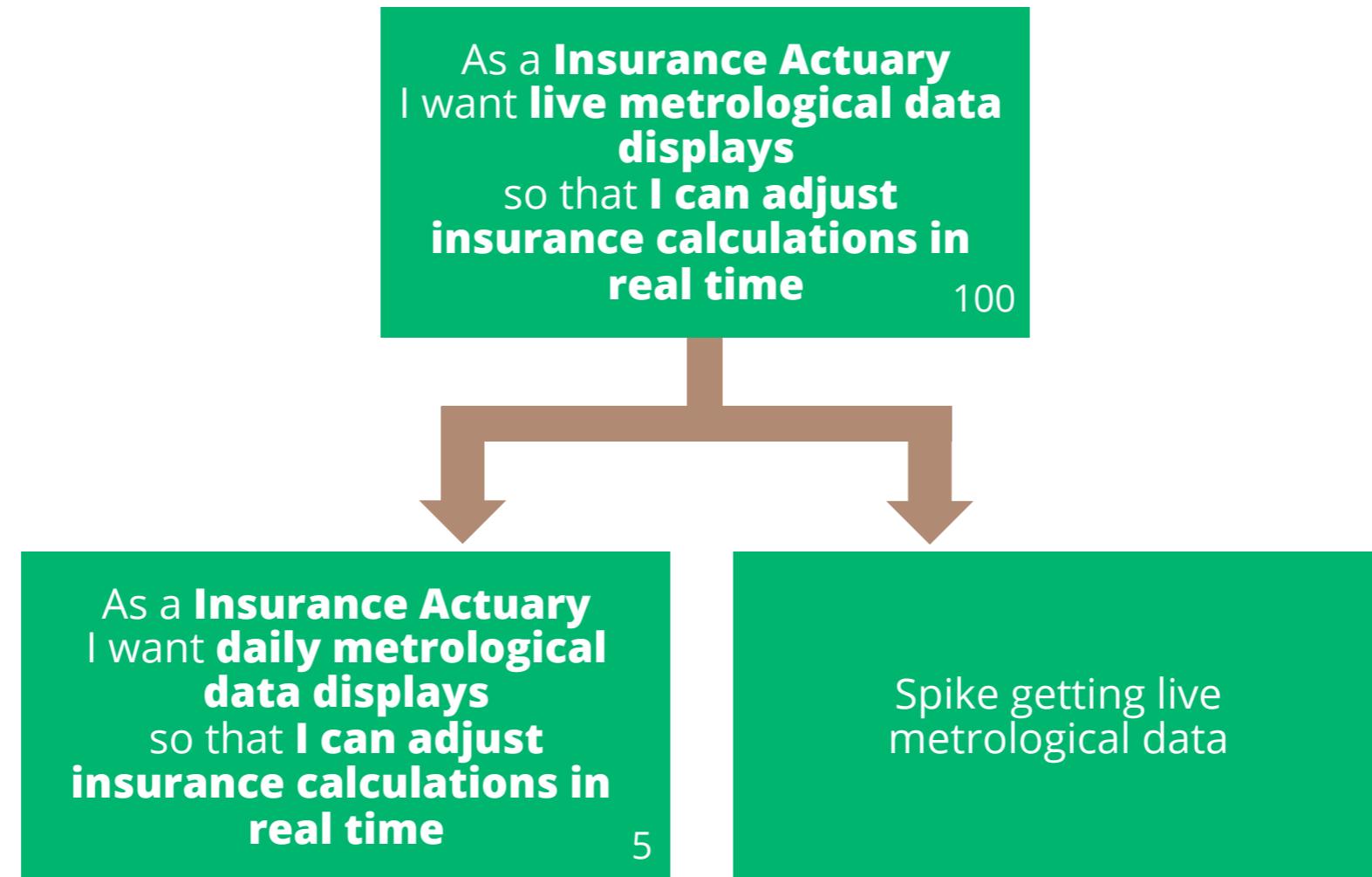


**Sign in with
LinkedIn**

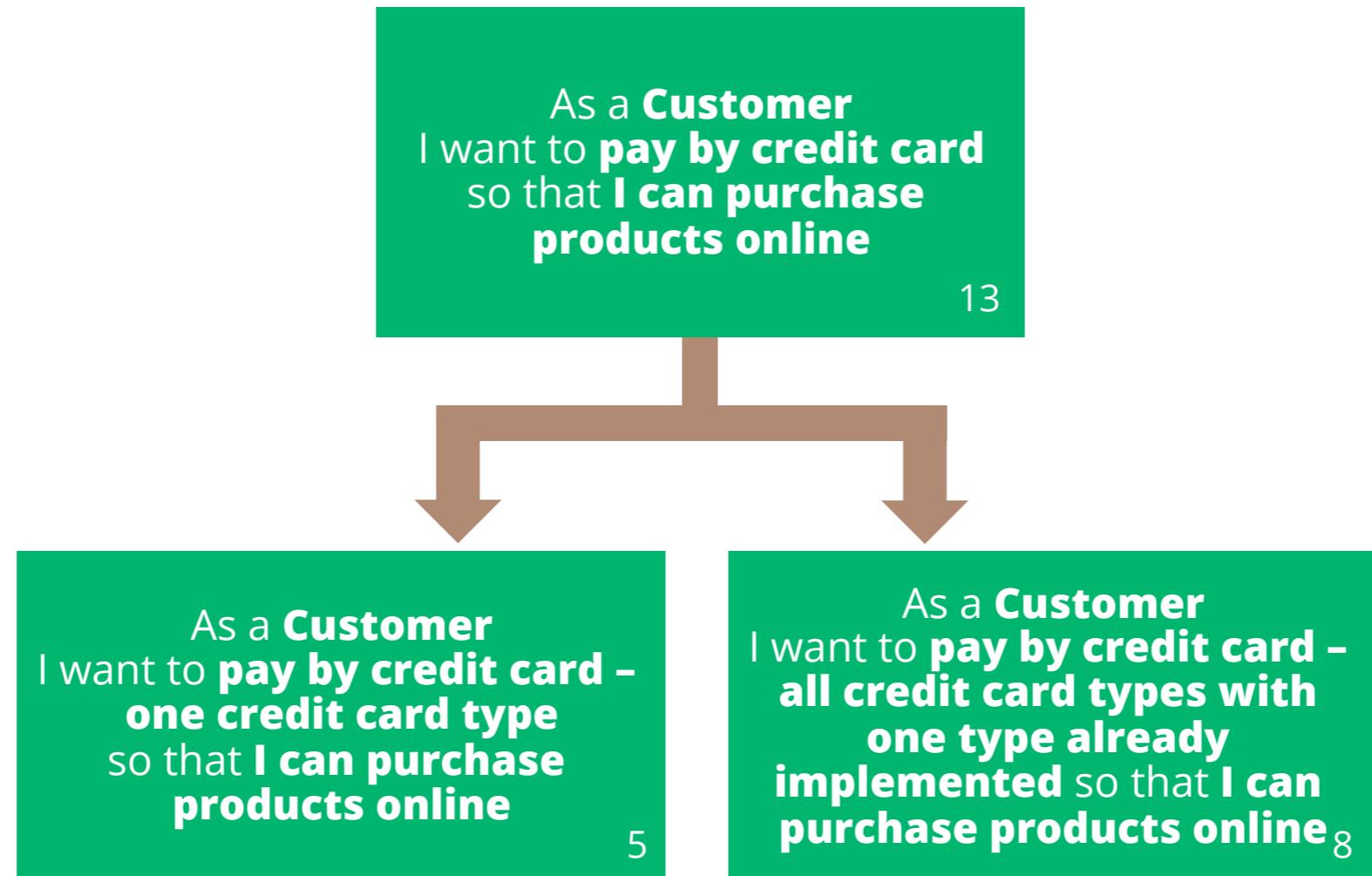
SPLITTING BY VALUE



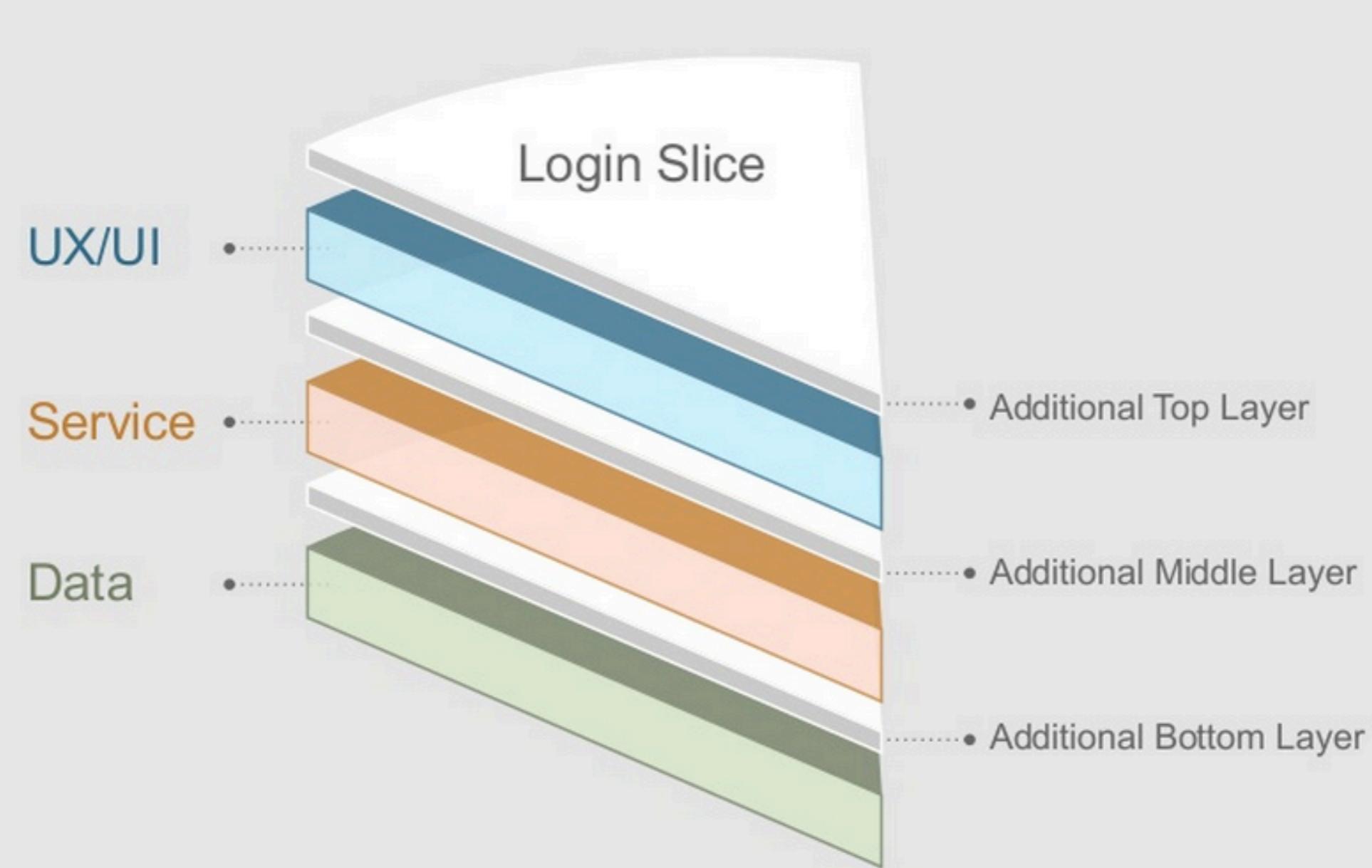
SPLITTING BY RISK



SPLITTING DEPENDENCIES



SLICING VERTICALLY



SLICING VERTICALLY



WORKING WITH PERSONAS

PERSONAS

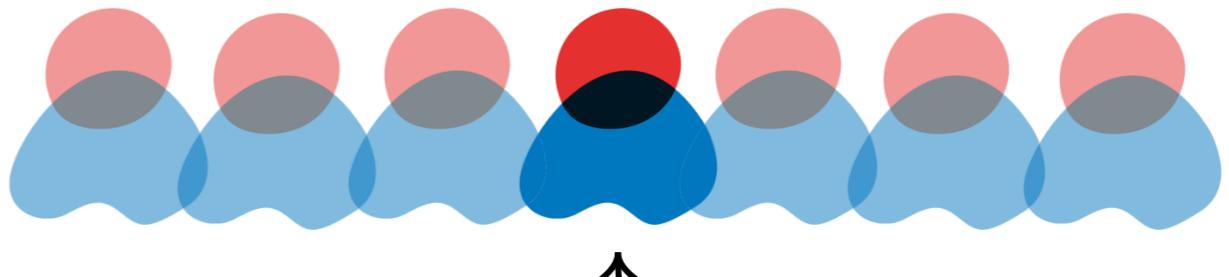
- Concrete characterization of a user group
- Archetype / representation / profile
- Description of end users based on research
- Reference document
- It can be used a synthesis of different profiles

HOW TO DEVELOP PERSONAS

1. Conduct research
2. Identify and cluster the behavioral variables
3. Identify trends
4. Create a persona for each of the trends
5. Create scenarios and tell stories
6. Create personas documentation

HOW TO DEVELOP PERSONAS

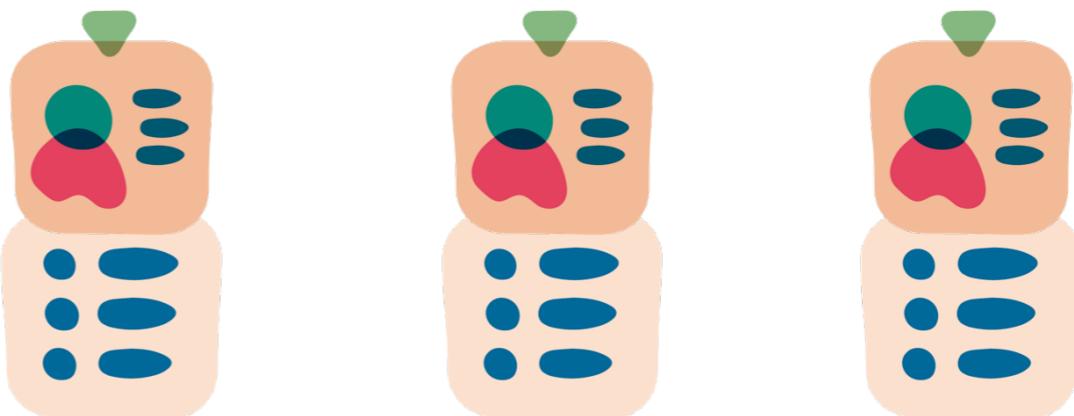
Identify roles and goals



Distribute behaviorally distinct traits



Assign attributes and create the personas based on research



<Persona name>



“Insert quote that characterises this persona in one sentence.”

 Lorem ipsum dolor sit amet, consectetur
 adipisicing elit, sed do eiusmod tempor
 incididunt ut labore et dolore magna aliqua. Ut
 enim ad minim veniam, quis nostrud
 exercitation ullamco laboris nisi ut aliquip ex
 ea commodo consequat. Duis aute irure dolor
 in reprehenderit in voluptate velit esse cillum
 dolore eu fugiat nulla pariatur.

Name <name>

Type <type>

Role <role>

Motivations

- Excep^teur sint occaecat cupidatat non.
- Sunt in culpa qui officia deserunt mollit anim id est laborum.
- Excep^teur sint occaecat cupidatat non.

Goals

- Excep^teur sint occaecat cupidatat non.
- Sunt in culpa qui officia deserunt mollit anim id est laborum.
- Excep^teur sint occaecat cupidatat non.

Pain points

- Excep^teur sint occaecat cupidatat non.
- Sunt in culpa qui officia deserunt mollit anim id est laborum.
- Excep^teur sint occaecat cupidatat non.

Common profiles

- Excep^teur sint occaecat cupidatat non.
- Sunt in culpa qui officia deserunt mollit anim id est laborum.
- Excep^teur sint occaecat cupidatat non.

Behaviours

Variable description



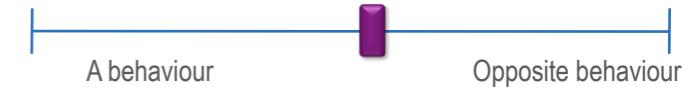
Variable description



Variable description



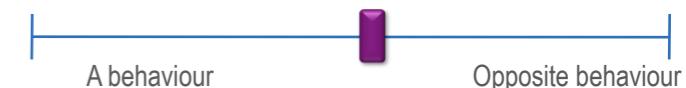
Variable description



Variable description



Variable description



ROLES VS. PERSONAS

- Personas can easily become a reflection of roles within a system,
- Personas should be able to identify different behaviors of roles
- Temptation to create role-based persona based on existing team knowledge of the system

QUESTIONS?

ThoughtWorks®



ESTIMATION

WHY ESTIMATE?

- Provides a measure of progress
- Helps with prioritization
- Estimation discussions are very helpful for defining an approach to delivering an individual story
- BUT:
 - Time consuming for large numbers of stories
 - Not a measure of time

WHAT TO ESTIMATE

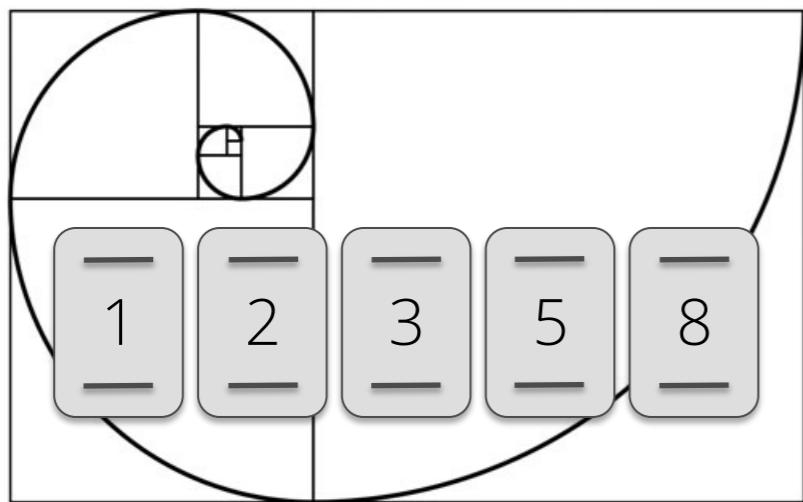
- Don't try and estimate all the stories in a project
- Have a prioritized backlog and estimate highest priority first
- Ensure there are estimated stories for the next Iteration
- Stories that are likely to be difficult

WHAT TO ESTIMATE

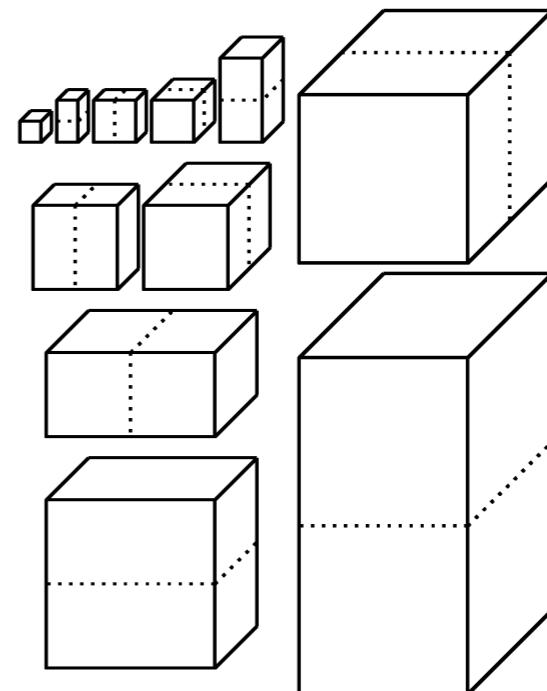
- A common view of what needs to be complete for a story to be counted as done and to include its story points in velocity
- Commonly includes: functional code completed, tests passed, acceptance test passed, and deployed to production environment
- Definition should be discussed and agreed as part of inception
- Should define a situation where the team will not have to work on the story again

HOW TO ESTIMATE

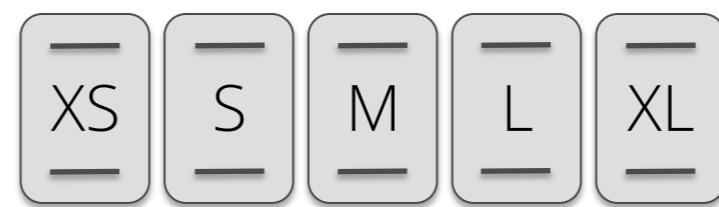
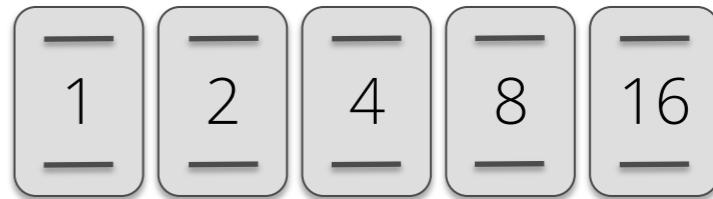
Fibonacci



Power of Two



T-Shirt size



GOOD ESTIMATION PRACTICES

- A story should fit in an iteration
- The smallest story should represent a single story point
- Delivery folks should participate in estimation
- Participants should discuss the story until they all agree on an estimate
- There should be sufficient understanding of how the story will be delivered
- Use spikes when technical uncertainty is too high
- Don't overdo it!

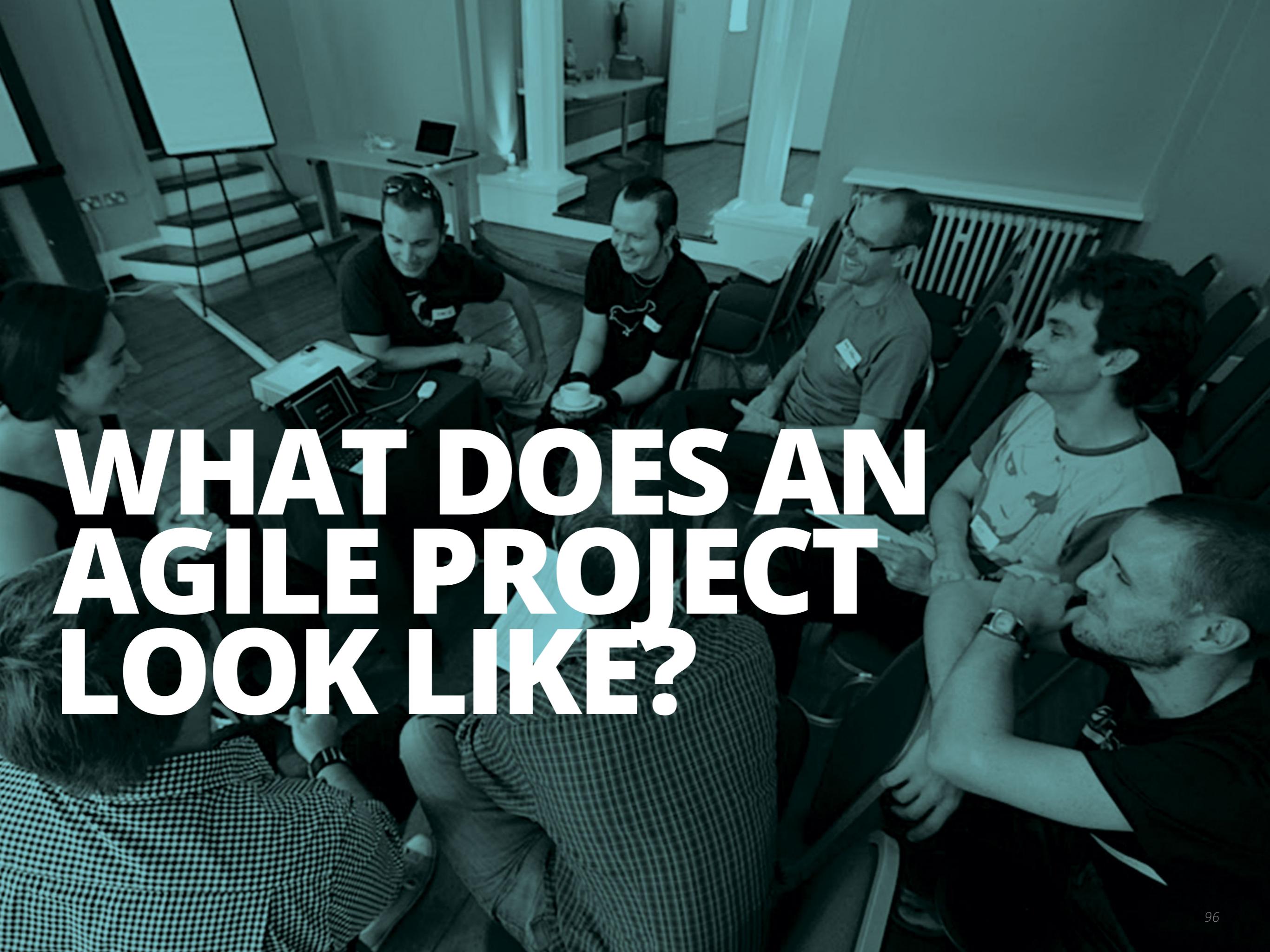
QUESTIONS?

ThoughtWorks®

EXERCISE

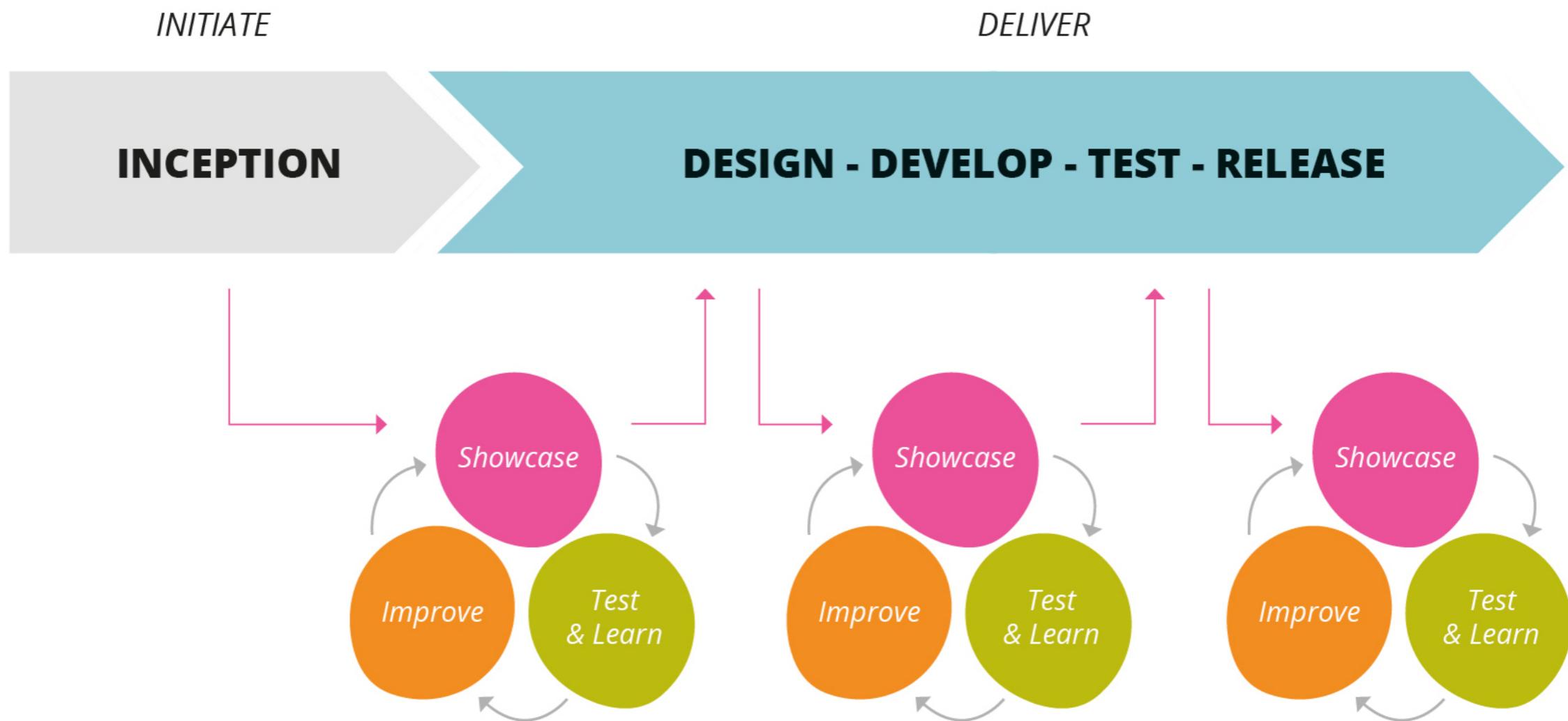
Doggy Planning



A black and white photograph showing a group of approximately ten people in an office environment. Some individuals are seated at desks, looking down at papers or computer monitors, while others are standing or walking through the space. The office has large windows, a staircase, and various pieces of furniture like desks and chairs. The overall atmosphere appears collaborative and busy.

WHAT DOES AN AGILE PROJECT LOOK LIKE?

WHAT DOES AN AGILE PROJECT LOOK LIKE



WHAT IS AN INCEPTION?



HOW INCEPTIONS FEEL

- Highly collaborative and inclusive
- Time-boxed and rapid, focused on doing 'just enough'
- Feedback-driven and highly adaptive
- Is not upfront analysis
- Highly visual and workshop oriented to help evolve a vision for the project.



INCEPTION OBJECTIVES

Build a shared understanding of project vision and goals

Evaluate high-level scope and core processes

Evaluate key risks, issues and constraints in delivering to the end goal

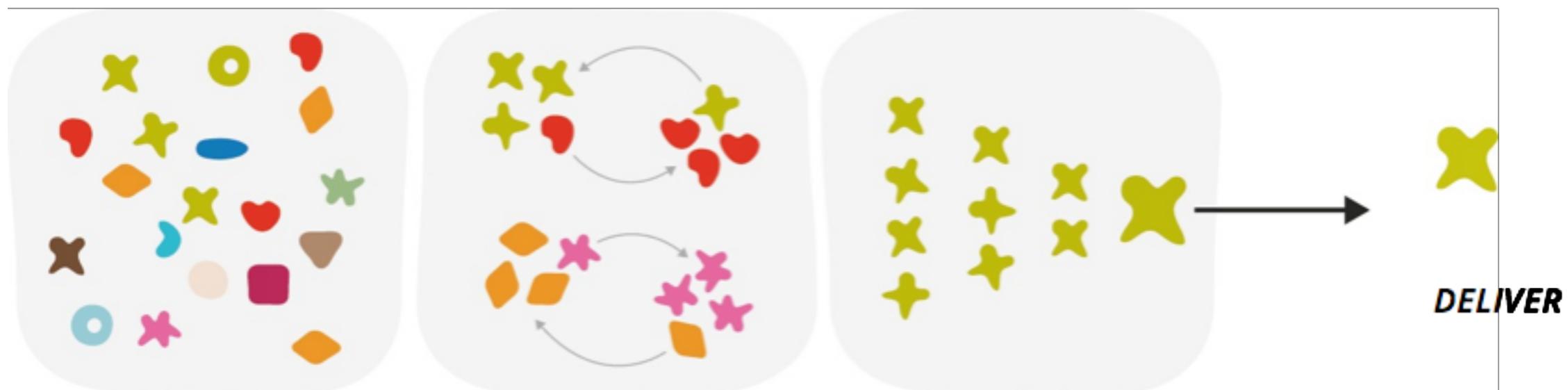
Establish technical and testing approach

FAST AND COLLABORATIVE

STAGE 1

STAGE 2

STAGE 3



OPEN

ORGANISE

FINALISE

OUR JOURNEY



OUTCOMES

- All artifacts necessary to get started
- A prioritized list of features that deliver the most business value
- Consensus about the future state
- A shared vision of the solution
- Jointly produced system requirements
- A realistic, achievable plan for an early release of business value.

WHAT WE'LL DO

together

ssing
ng

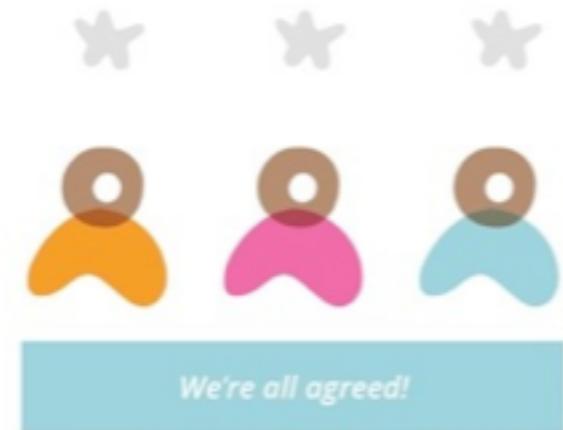
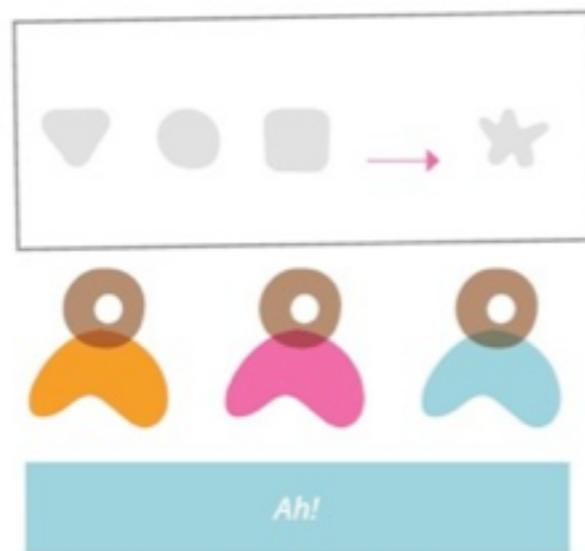
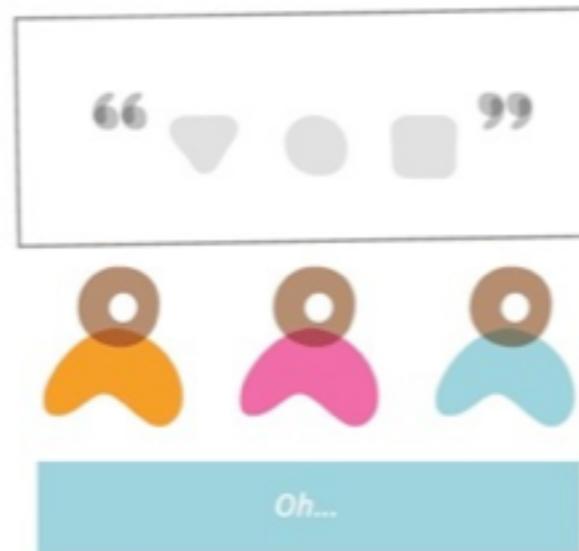
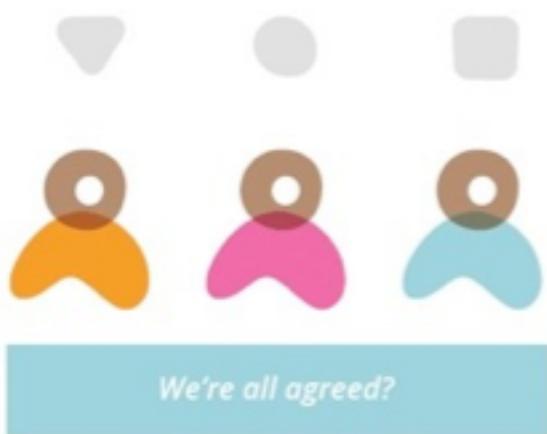
whole

ther

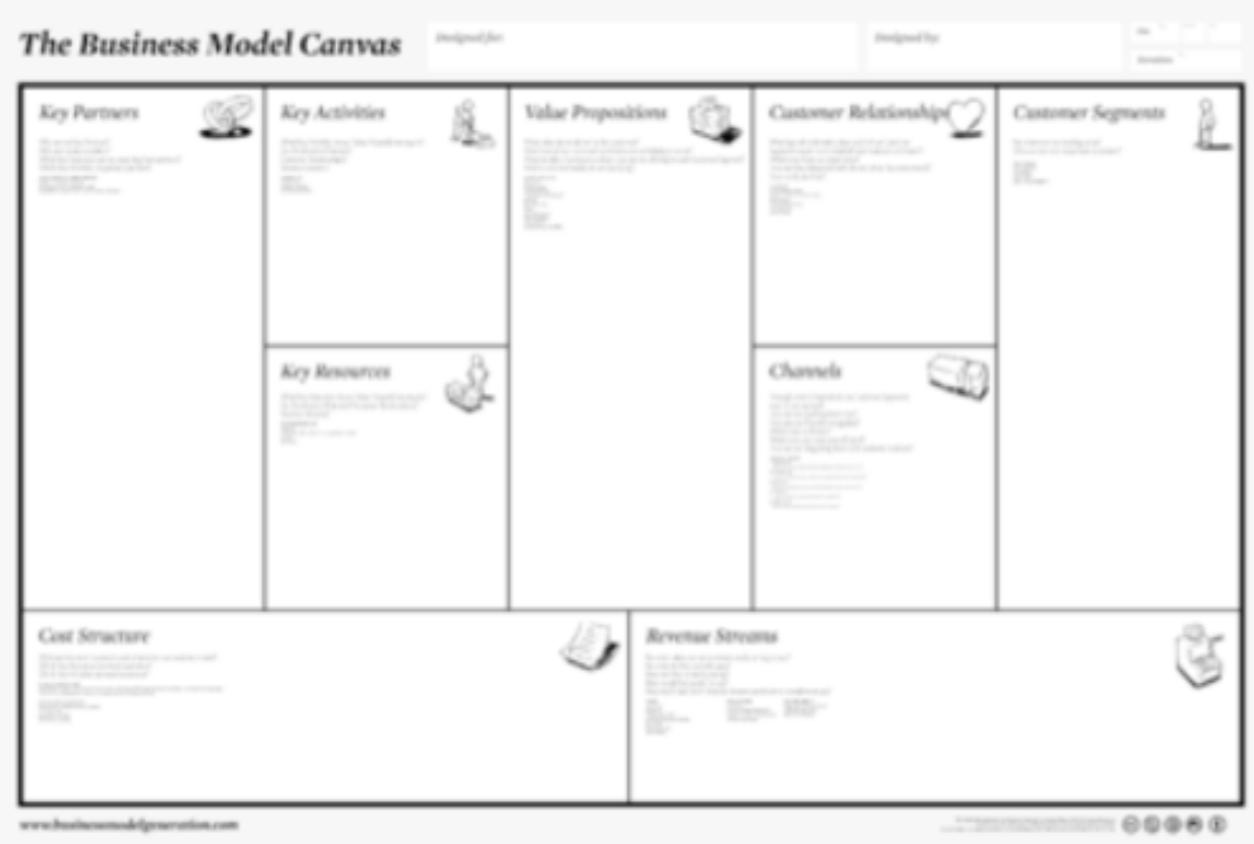
Integrate often

4/11/11

CREATE A SHARED VISION



UNDERSTAND BUSINESS VISION



USER JOURNEYS AND WHITEBOARDING

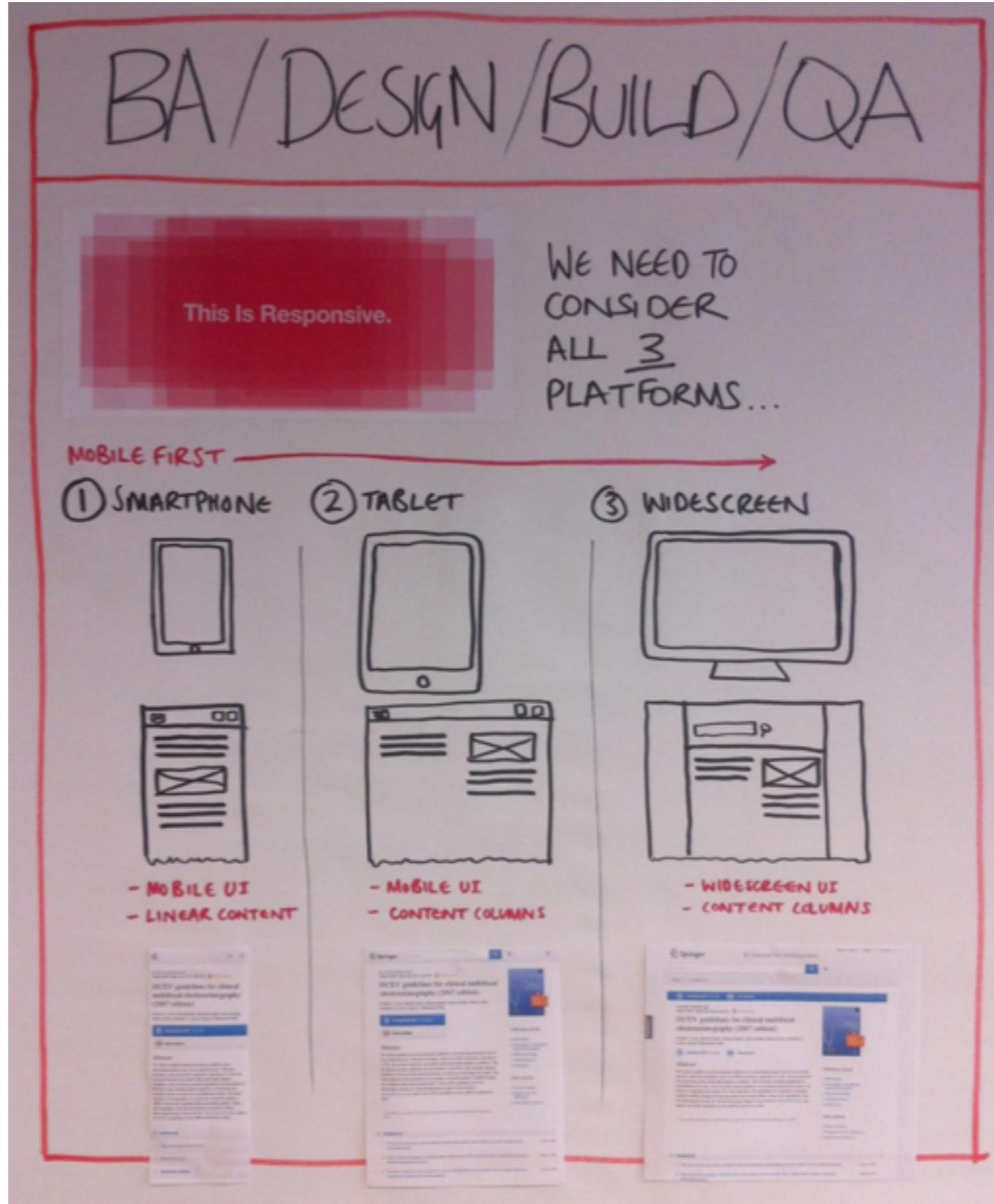


USER EXPERIENCE DESIGN

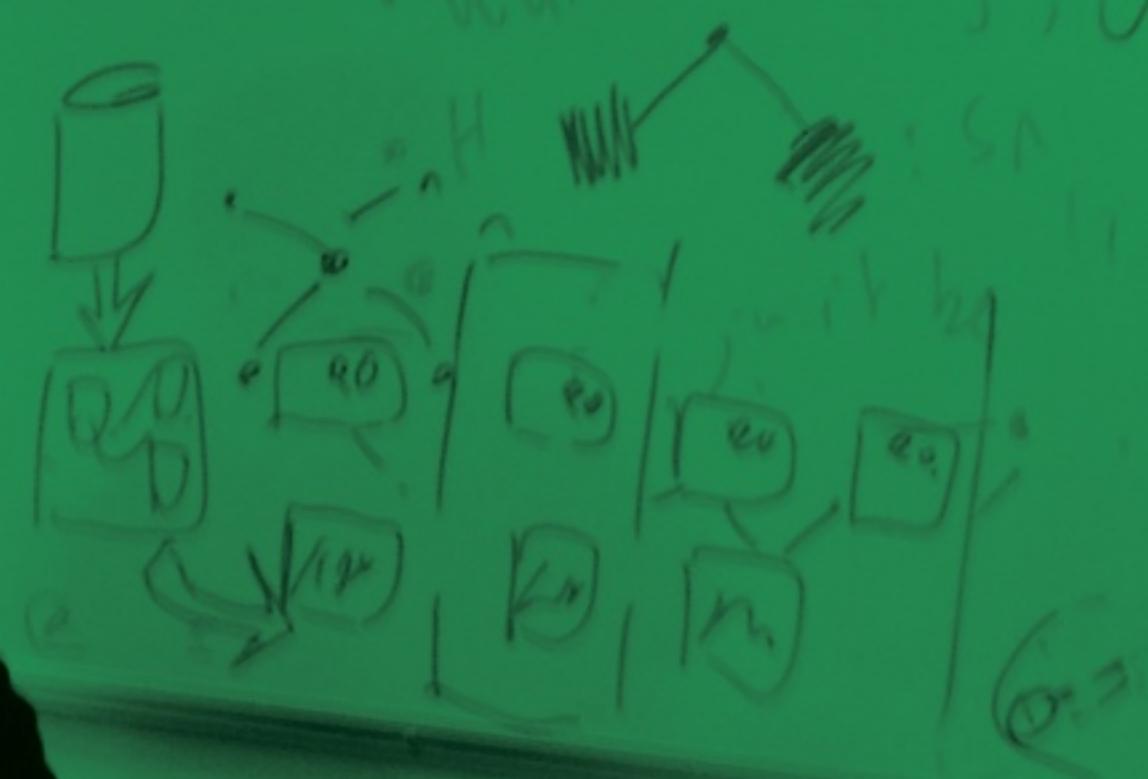
- Discovery
 - User Personas
- Concept
 - User Journey
 - Information Architecture
 - Collaborative Design
 - Wireframes
- Visual Design



COLLABORATIVE DESIGN APPROACH

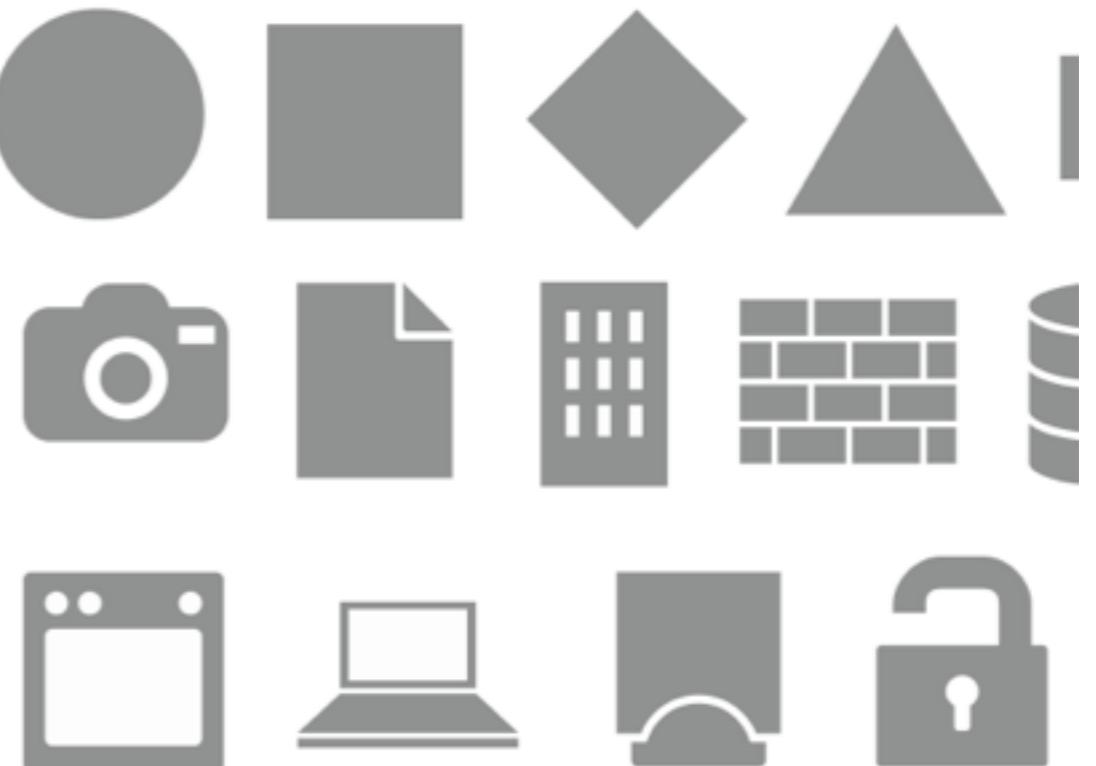


UNDERSTAND AND PROPOSE TECHNICAL SOLUTION



TECHNICAL APPROACH

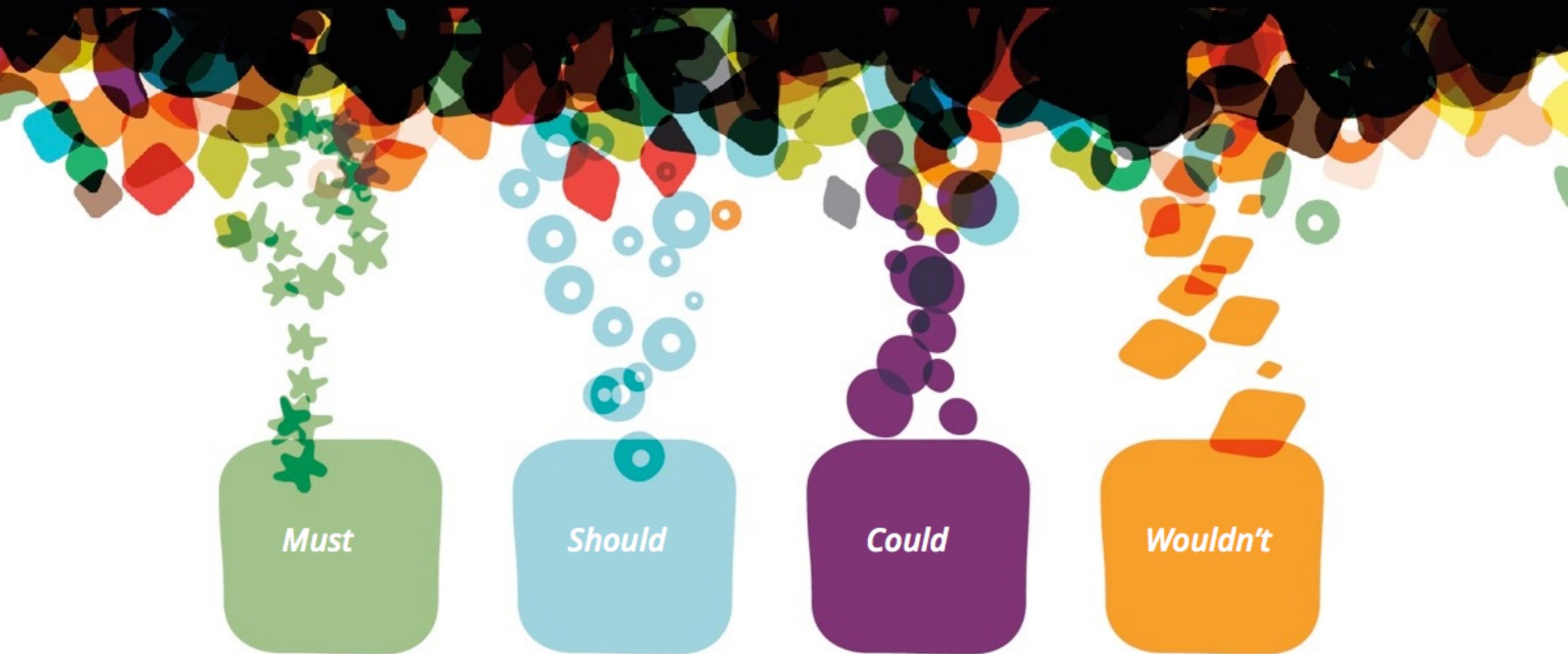
- Technical visioning
- Architecture
- Proposed testing approach
- Addressing Non Functional Requirements (NFRs)



IDENTIFY & DISCUSS FEATURES AND STORIES



PRIORITISE





ESTIMATION

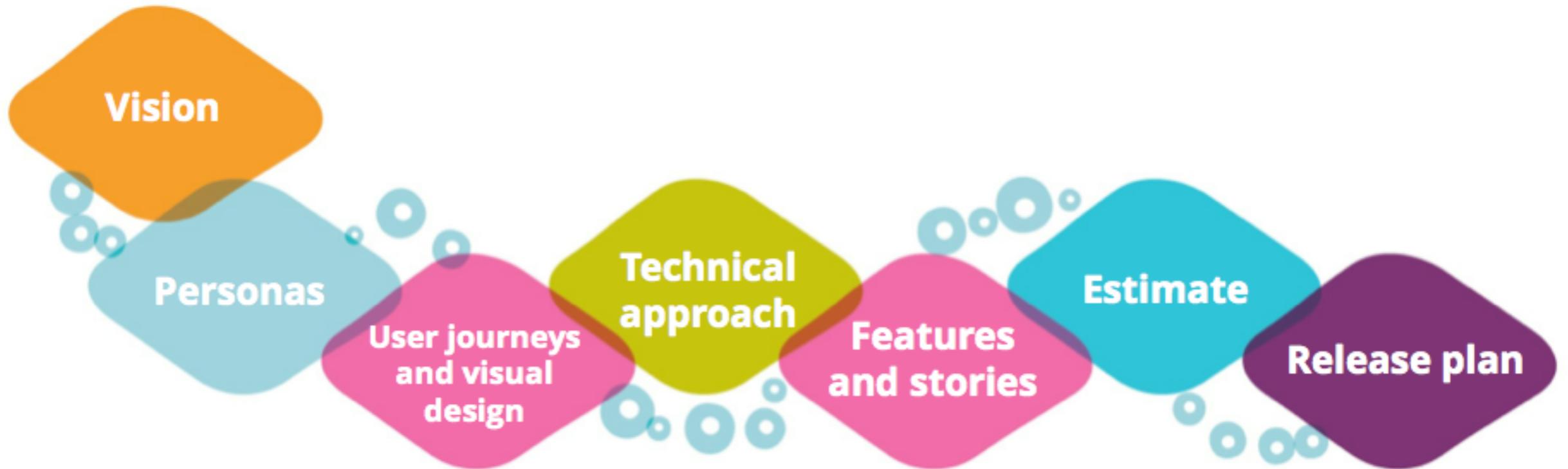
We capture...

RISKS
ASSUMPTIONS
ISSUES
DEPENDANCIES

PREPARE RELEASE PLAN



QUICK RECAP



A photograph of three people in an office setting. A woman with glasses and a scarf stands on the left, looking down at a piece of paper. A man in a grey sweatshirt is crouching in the center, also looking at the paper. Another man on the right is smiling and laughing. The background shows walls covered in numerous sticky notes.

WHAT TO EXPECT

INCEPTION DELIVERABLES

- Business Vision and Project Objectives
- Prioritised High Level Requirements (Stories & NFRs)
- Technical Approach
- Visual Design Mock-ups
- Release Plan
- RAIDs & Communication Plan
- Next Steps

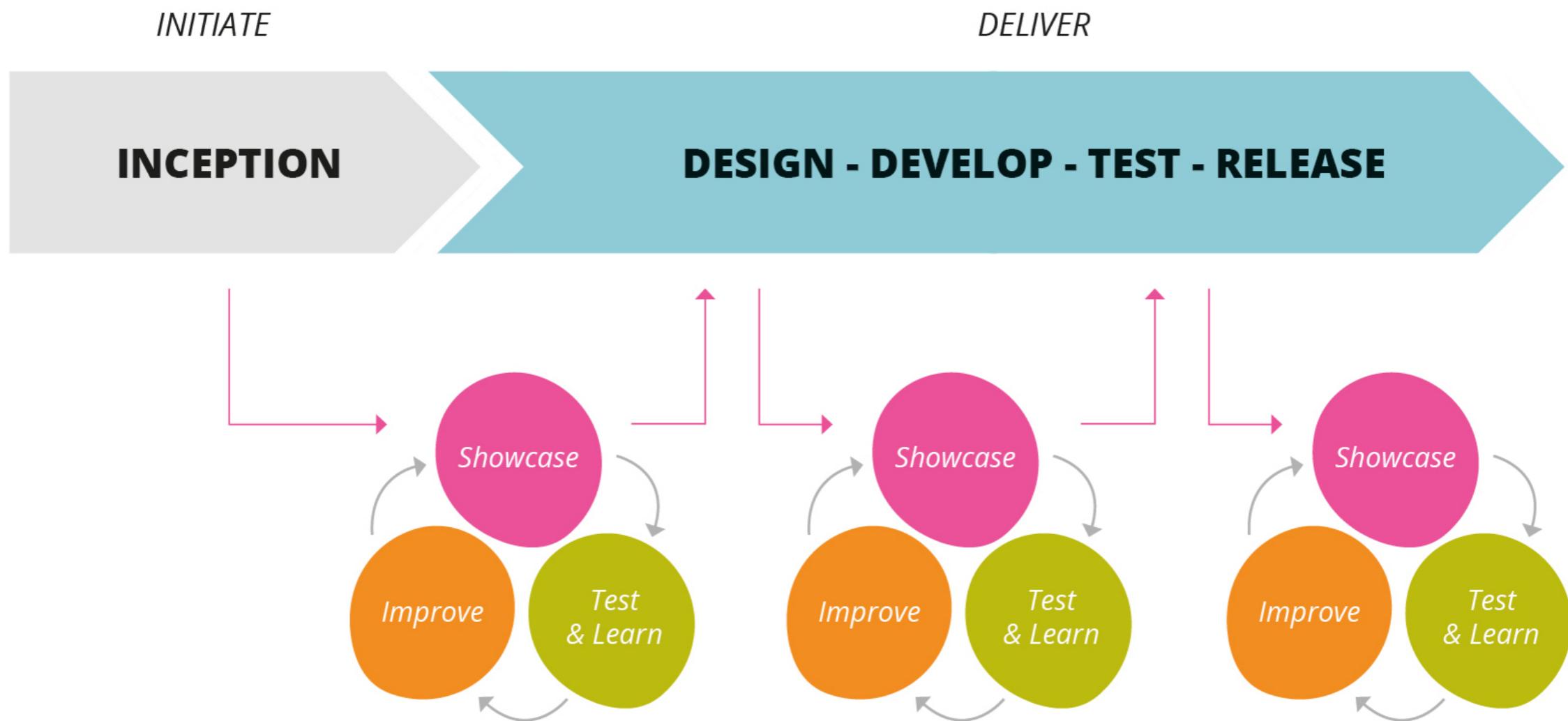
QUESTIONS?

ThoughtWorks®



LET'S GET TO DELIVERY

WHAT DOES AN AGILE PROJECT LOOK LIKE



As a **customer**

I want to **be able to sign in using**
my Facebook account

so that **I don't need to remember**
***another* username and**

password

INITIAL REQUIREMENTS



project inception

*unfinished from
previous projects*

ad-hoc conversations

analysis activities

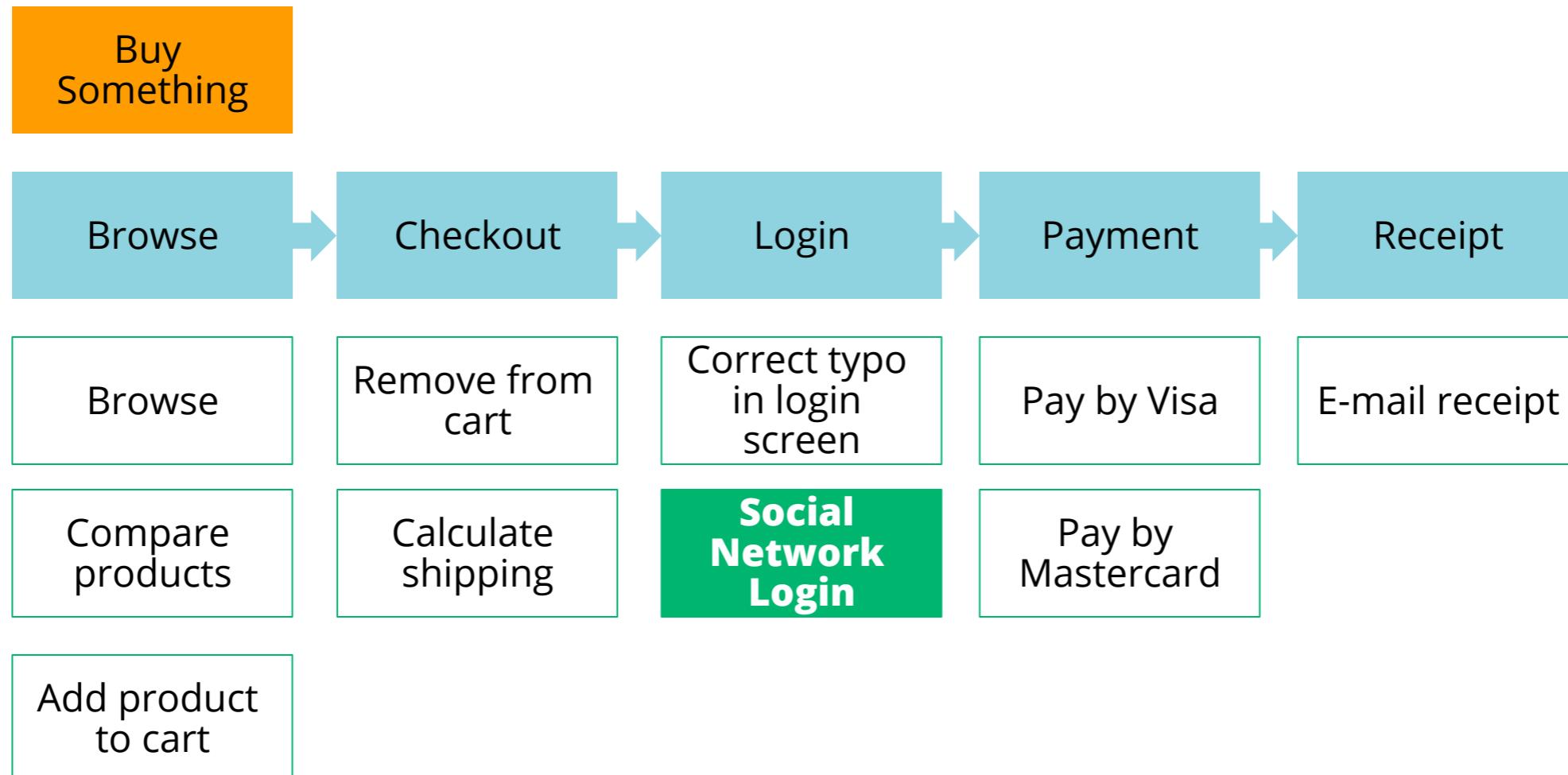
non-functional reqs

- Initial project requirements come from a wide variety of sources. Perhaps the most common source is inception workshops that involve the core delivery team and stakeholders
- Requirements will often vary greatly in level of details ... from loosely-defined **epics** through to **stories**, down to very well-defined **bugs**

PRODUCT BACKLOG

- These initial requirements form our Product Backlog
- At the simplest, just a prioritized/ranked list of requirements
- The Product Backlog is owned and maintained by the Product Owners with assistance of Analysts over the lifespan of the project
- Product Backlogs evolve over the course of the project (they are not fixed after being defined up front!)

STORY MAP



A STORY MAP ON THE WALL



NON-FUNCTIONAL REQUIREMENTS

some examples:

Auditability	Authentication	Authorization
Availability	Help	Localization
Performance	Portability	Regulatory
Scalability	Security	Usability

NFR: WHAT TO DO WITH THEM?

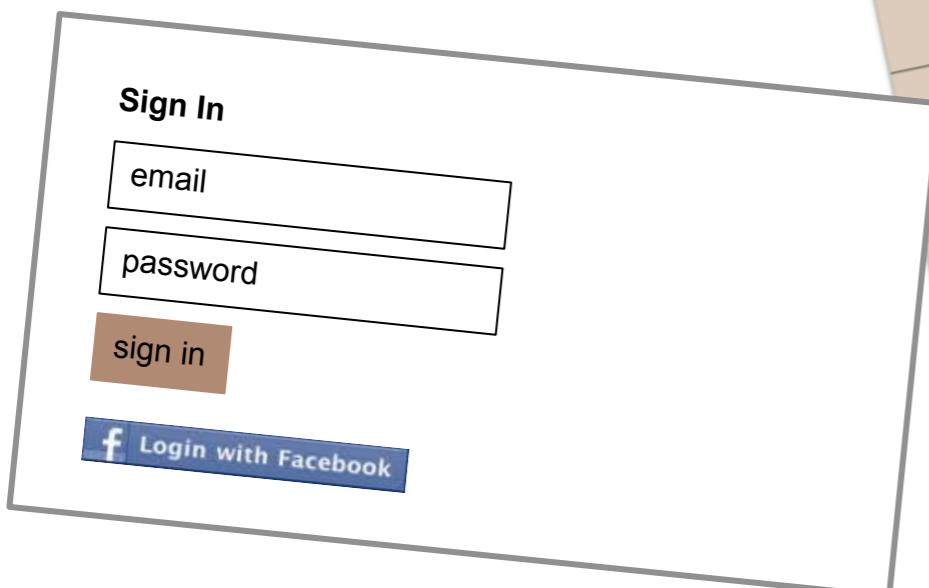
- Discover which ones matter
- New stories (a mind map may help)
- Acceptance criteria
 - in some stories
 - in all stories
- Constraints
 - “Constraint cards”
- Review the list of ‘ilities’ in a facilitated session

ANALYSIS AND UX ELABORATION

- Analysts, UXers and Product Owners progressively elaborate Stories for development – high priority first!
- Stories are picked up for elaboration far enough ahead of when they are expected to be implemented to ensure detail is ready, but not so far in advance that it's hard to change course
- Elaboration activities include:
 - Story breakdown
 - Finishing User Stories, Acceptance Criteria and Narratives
 - Prioritization
 - Estimation

ANALYSIS AND UX ELABORATION

Sign in with Facebook



Sign in with Facebook

As a customer I want to be able to sign in using my Facebook account so that I don't need to remember *another* username and password

Narrative

Requested by our customers.
Our facebook app ID is 9873948719348

Acceptance Criteria

- Should capture Facebook name, email, **country** and birthday
- Should be available anywhere in site

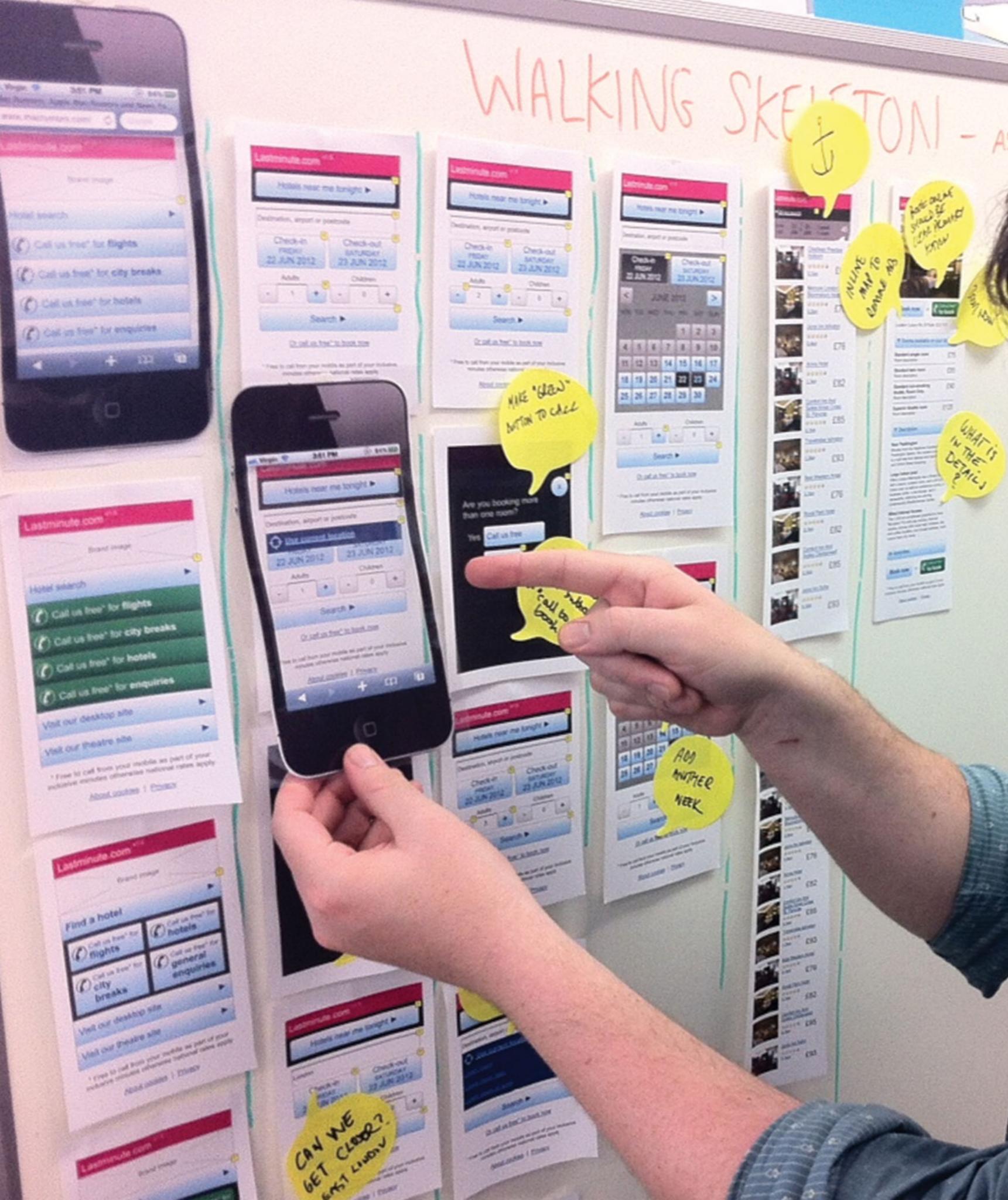
ThoughtWorks®



LOGO	13/6/14
NAME (ACQUISITION)	
PRICE	PRIVILEGED NON-PRIVILEGED
<input checked="" type="checkbox"/> COVERED	<input type="checkbox"/> NOT COVERED
BEST FOR	



WALKING SKELETON - AS IT IS



ESTIMATION AND PRIORITIZATION

- Periodically (every couple of iterations) prioritization should be revisited with Product Owners to see if any priorities have shifted
- Similarly if elaboration identifies substantial changes in scope or potential effort, estimates should be revisited with the Core Team
- This activity of revisiting the backlog for estimation, prioritization and add/remove stories based on recent learning is sometimes known as “backlog grooming”

ITERATION PLANNING MEETING

- Agenda
 - Confirm priorities
 - Calculate expected team capacity
 - Overview of high priority stories
 - Development team indicates which stories they can commit to that iteration

STAND-UP MEETING

- Team meets for 15 minutes daily to provide a status update among team members:
 - What did I accomplish yesterday?
 - What will I do today?
 - What obstacles are impeding my progress?
- Goal is to highlight potential challenges and identify ownership to coordinate efforts to resolve difficult or time-consuming issues
- It is not a goal to resolve those issues in this 15 minute meeting

Yesterday I finished up the logout story, today I'm hoping to pick up the Facebook login story. Tony and Jasmine are you available after standup to kick it off?

Yep.

Sure thing.

Thanks.

Yesterday we integrated the Facebook JS library for the
'facebook share' feature.

Eric: let us know if you have any questions.

Jane: we've a problem with UAT environment DB.

Okay, thanks for the heads up.
Let's take this offline.



STORY KICK-OFF

- Product Owner or Analyst introduces the rest of the team to the story and answers any questions that come up at this point (may tweak story details)
- Developers describe how they intend to approach the solution
- Testers describe how they intend to approach the testing
- Designers talk about how the UX should be like
- Meeting ends when everybody is comfortable (5-30min)

DEVELOPMENT

- Developers work alongside UX, Analyst, Product Owners and Testers as required. Here conversations fill in any remaining details and any options (such as different solutions) are evaluated
- Testers may also prepare test cases and annotate on the story
 - Test cases are not acceptance criteria!
 - Test cases will include things not covered in acceptance criteria
- Development continues until developers are confident the acceptance criteria are met, and an environment can be provided for testing
- Do regular desk checks with the rest of the team

HANDOVER TO TEST

- Developer hands over story to Tester
 - If available, analyst, designers and product owners might be included
- Demonstrates functionality against the acceptance criteria in the test environment (if can't be demonstrated... it isn't ready!) (5-15 min)

TESTING

- Tester finalizes and executes test cases for the story.
Will seek additional clarification from Product Owners, Analysts, Designers or Developers
- Tester will do further exploratory testing to test the edge cases of the solution
- Any issues identified that would prevent the story from meeting acceptance criteria should be properly handled

STORY ACCEPTANCE

- Analyst or Tester demonstrates the solution to Product Owners to answer two questions:
- Does this solution meet the acceptance criteria (as agreed)... are we done on this story?
- Based on now seeing the working software, does this inspire other changes to this feature?
- Demonstration often involves walkthrough of Given...When... Then acceptance criteria, as well as some exploratory testing by Product Owners
- It should not be the first time the Product Owners see the story implemented

SHOWCASE

- Product Owners and Team demonstrate the stories that have been accepted as 'done' during the iteration to the full set of users and stakeholders
ie. everyone with any stake in the project
- Feedback is gathered
All changes, ideas → new user stories (product owners prioritize)

RELEASE

- The earlier the product under development reaches the hands of the end-users the better – it provides an opportunity to measure and respond to real usage of the system while there is still time to adjust

RETROSPECTIVE

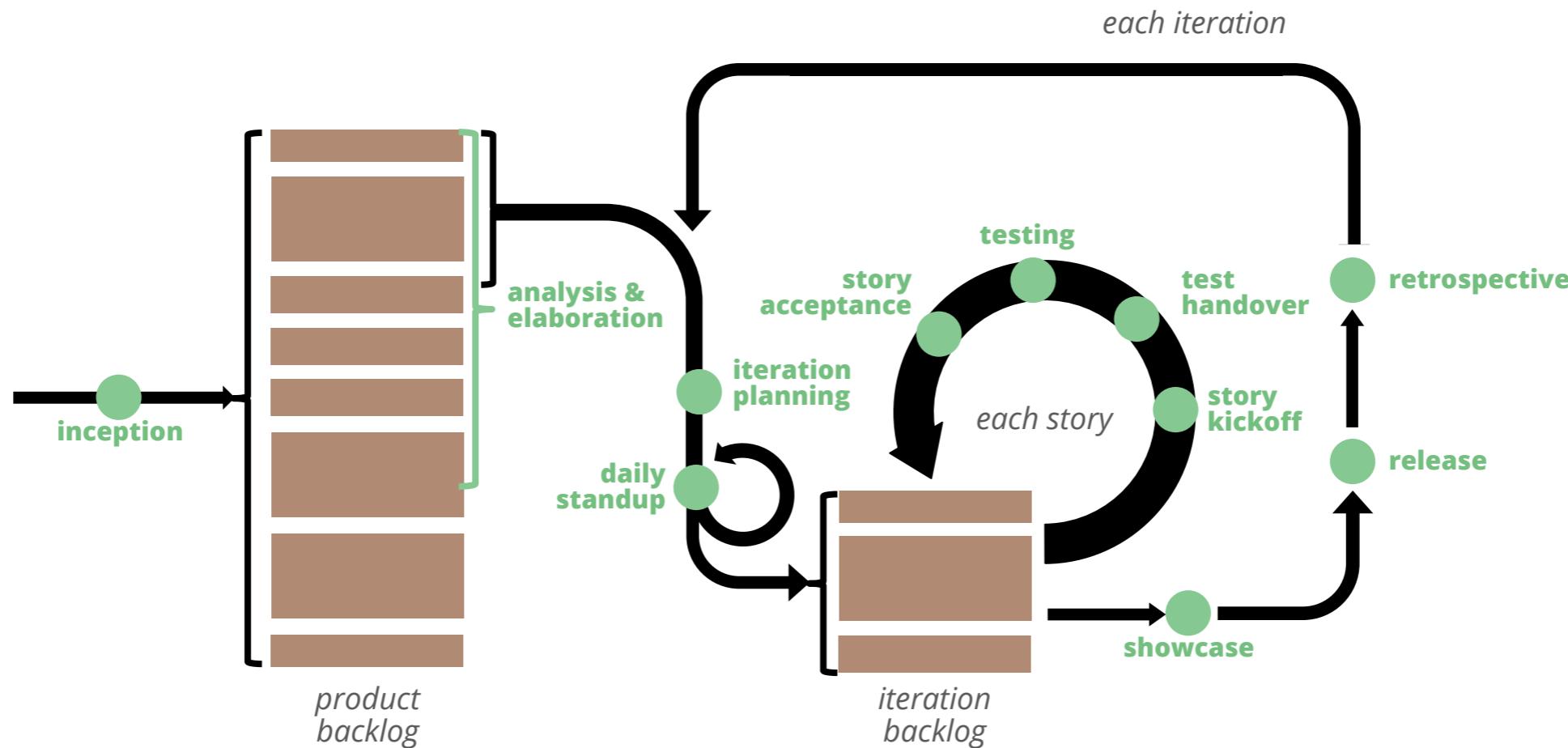
- Team brainstorms on sticky-notes:
 - What went **well** with the delivery process?
 - What went **not so well** with the delivery process?
 - What **ideas** do we have about things we can do differently?
 - What still **puzzles** us about the delivery process?
- Similar thoughts are combined into groups (sticky-notes placed together)
- Team votes on the groups to identify which they want to discuss most
- Team discusses top-voted groups, identifies and formulates actions: **who** will do **what** by **when**

RETROSPECTIVE PRIME DIRECTIVE

*Regardless of what we discover,
we understand and truly believe
that everyone did the best job
they could, given what they knew
at the time, their skills and
abilities, the resources available,
and the situation at hand.*

(Norm Kerth)

WRAPPING UP THE ITERATION



QUESTIONS?

ThoughtWorks®



PLANNING AND TRACKING

STORY WALL

Overview Kaizen **Kanban** My Work Project Backlog All History Project admin

Add Add with detail... ?

Select tree: **None** View as: **List** **Grid**

Group by: Story Status Color and Sort by: Priority Lane headings: Average Estimate – Planning Manage trees

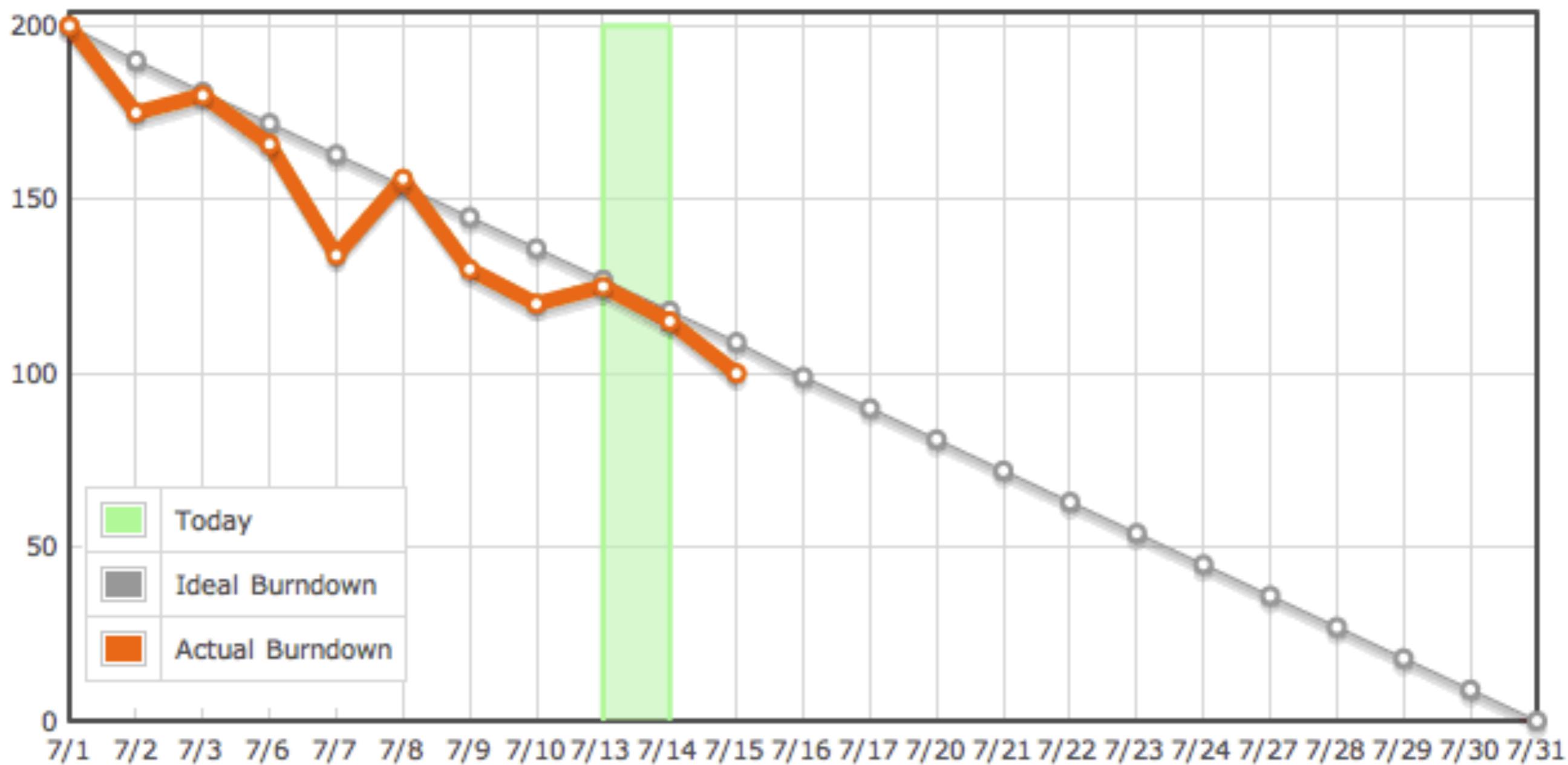
[Link to this page](#) [Add / remove lanes](#)

Ready for Analysis (2)	In Analysis (4)	Ready for Development (3)	In Development (3)	Ready for Testing (2)	In Testing (4)	Ready for Signoff (2)	Accepted (3.08)	Deployed (6)
Subscribe #86 to RSS feed	View new #51 mails differently so that I can easily scan	Merge #85 contacts Export my #84 contacts to iCal	Export my #83 contacts to excel View list #53 of mails User Session Timeout #3	Add e-mail addresses to addressbook from received Create Website Profile #1	Navigate #52 to and read e-mail	Save e-mail addresses in an online address book View #46 confirmation that e-mail was sent	Compose #49 and save (as a draft) a new mail Save sent #45 messages in an online 'sent items' box Specify #41 signature for all outgoing mail Reply to #40 email Delete #38 mail (single and multiple) Upload or #16 Replace Customer Holding Records Create #14	Pagination #50 through mails in folder to view more than 25 mails Send mail #48 Change #42 password

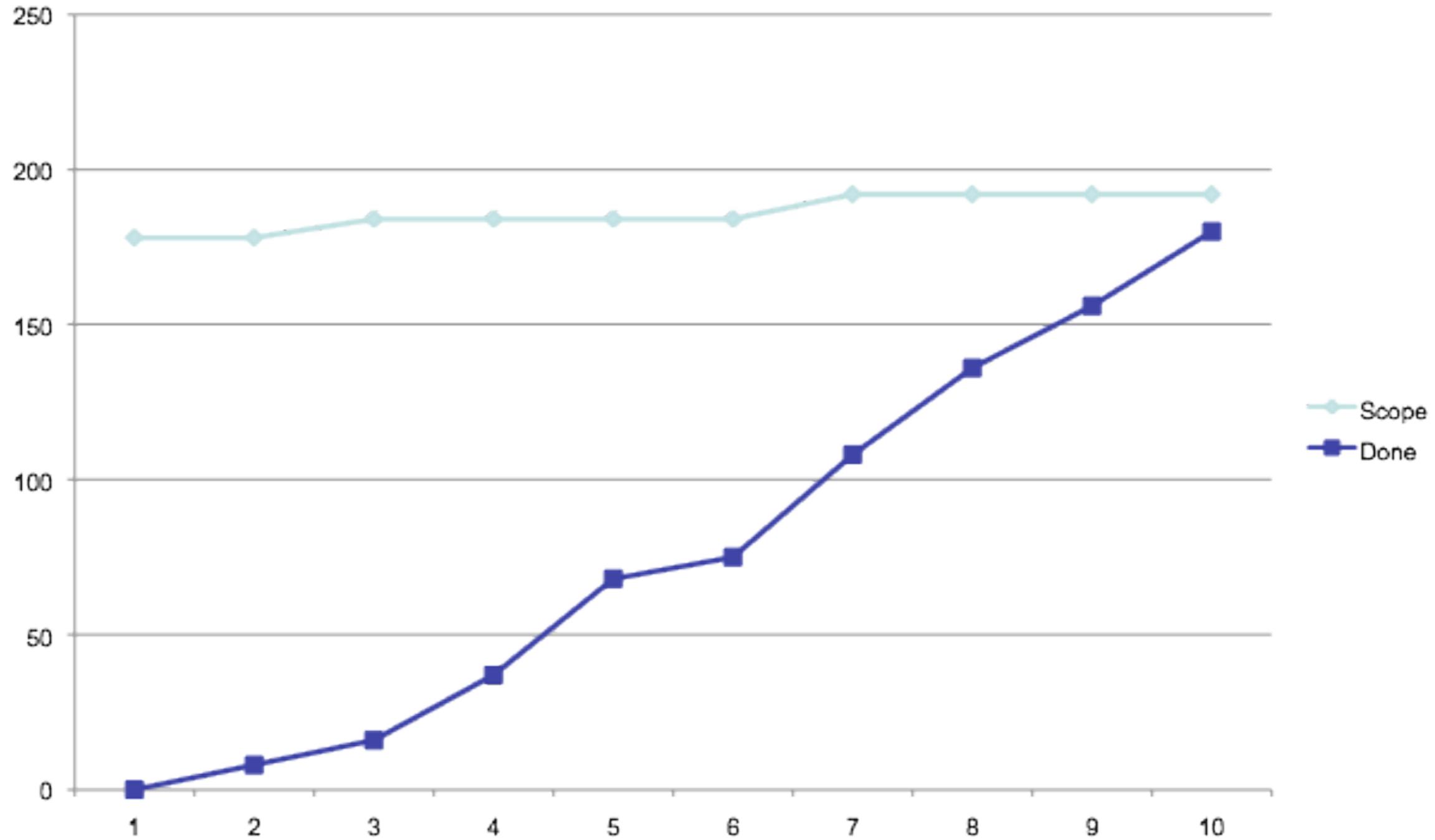


BURN DOWN

Daily Burndown



BURN UP



MANAGING RISKS

Overview BA Card Wall Development Card Wall Feature Tree QA Card Wall Risks

Release ▾ Add Add with detail ?

Select tree: None View as: List Grid

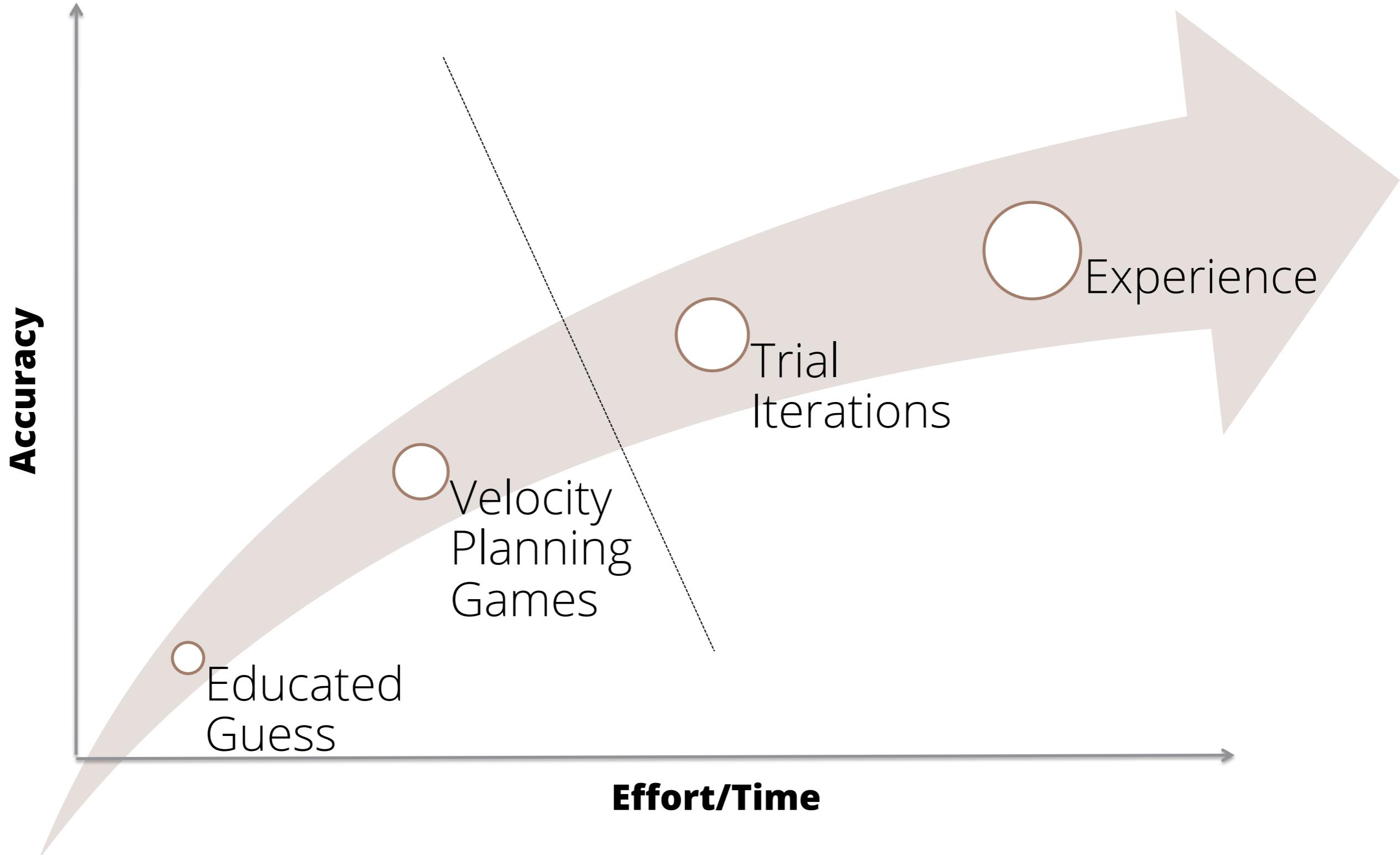
For selected: Edit properties Tag Delete More actions... Go Manage trees

Select: All | None Listed below: 1 to 5 of 5.

Print Link to this page Add / remove columns

#	Name	Action Due By	Risk Status	Risk Impact	Risk Likelihood	Risk group
111	Legacy system - dependency on client dev team	15 Mar 2008	In Progress	500000	High	Technical
110	Industry accreditation standards changing	15 Mar 2008	In Progress	750000	High	Accreditation
109	Integration with third party systems	15 Mar 2008	In Progress	25000	Medium	Technical
108	Lack of resources with required skillsets	15 Mar 2008	In Progress	54400	High	Resources
107	VPN connectivity to access client databases and systems	15 Mar 2008	Closed	12333	Medium	Infrastructure

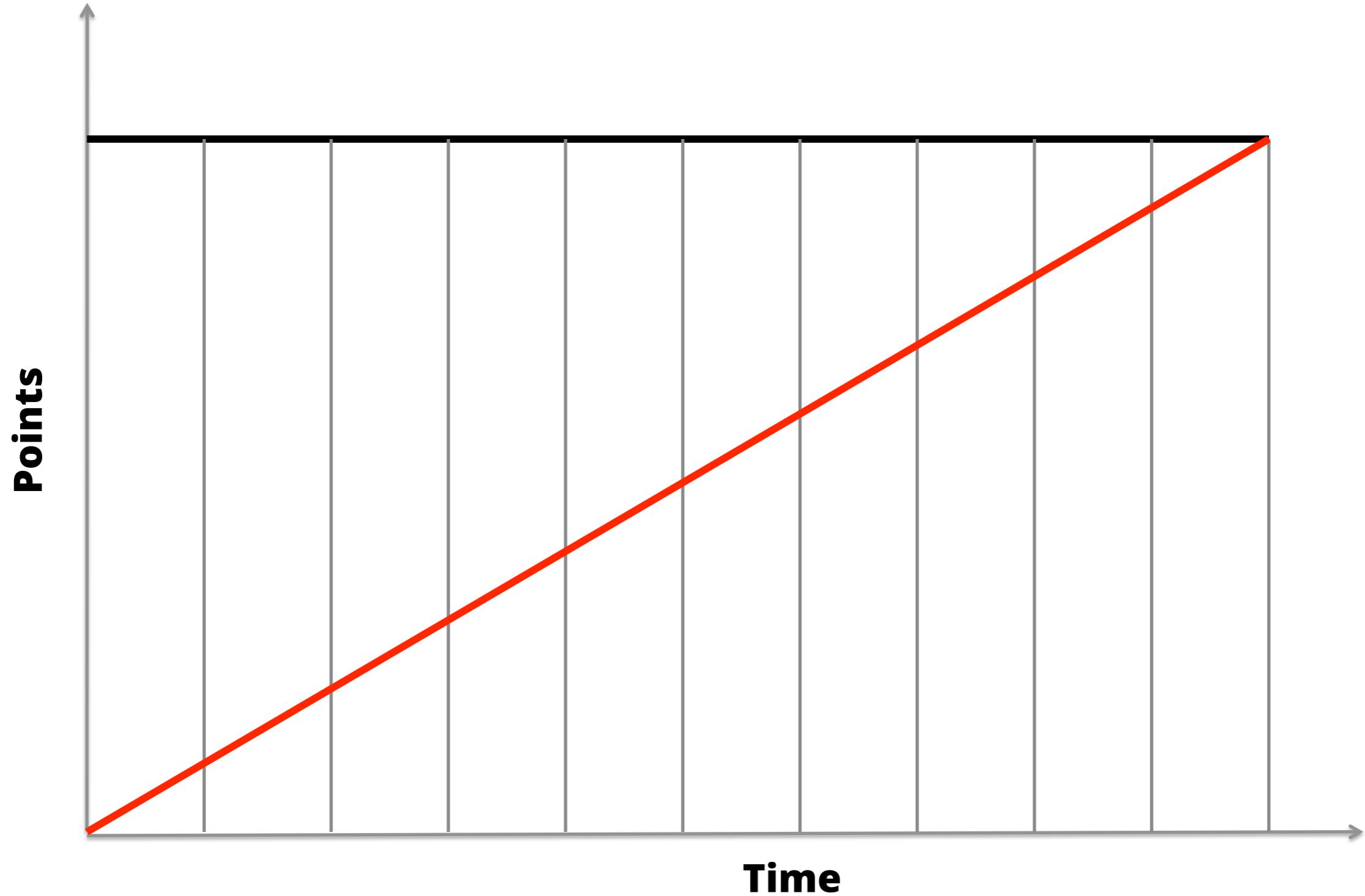
HOW TO DERIVE A PLANNING VELOCITY



HOW LONG WILL THE PROJECT TAKE?

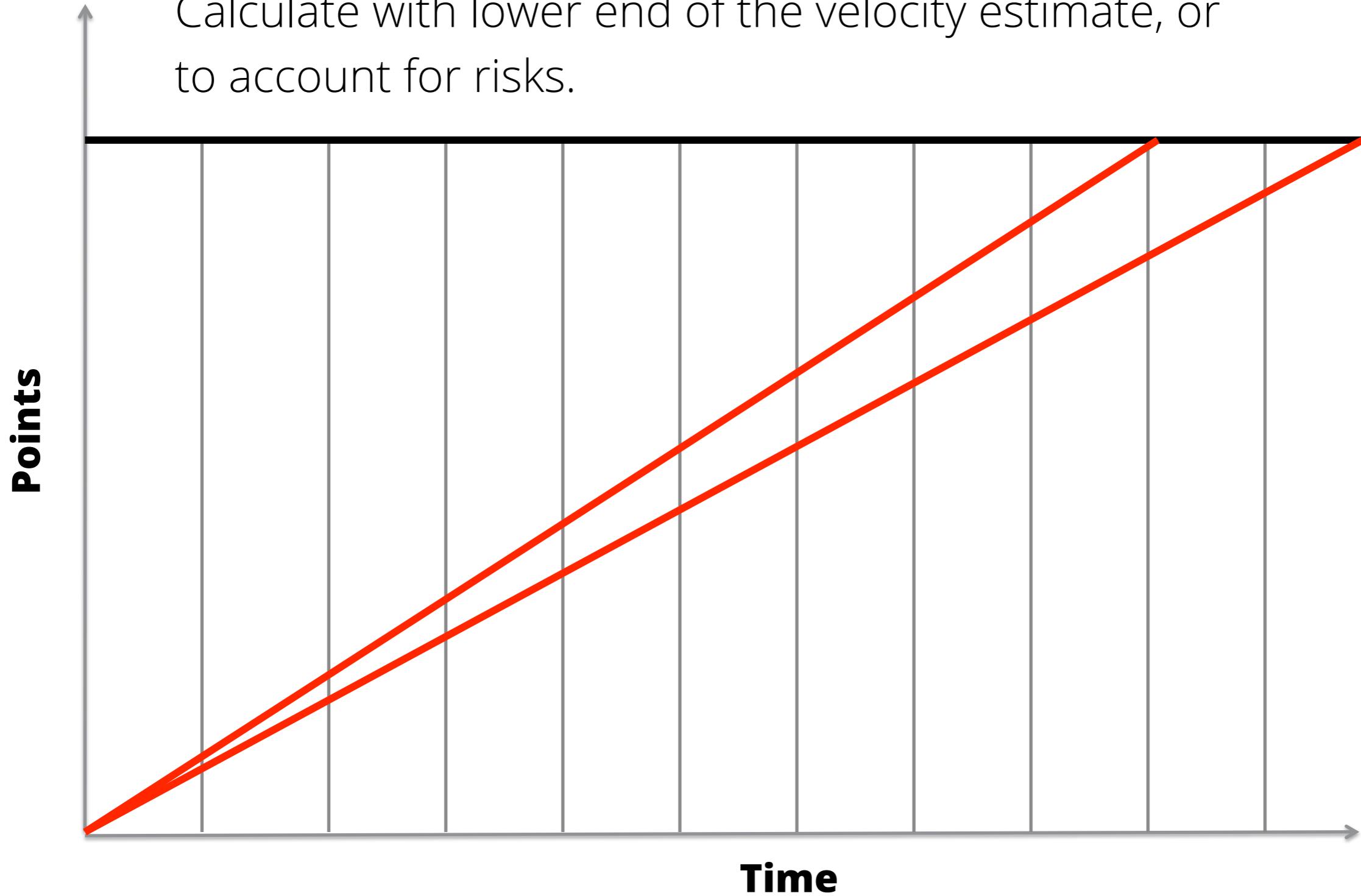
1. Run Velocity Planning games
2. Find the total number of story points
3. Find raw velocity per iteration
estimated velocity per pair X steady state pairs
4. Find number of iterations
total number of story points / estimated velocity

RAW VELOCITY



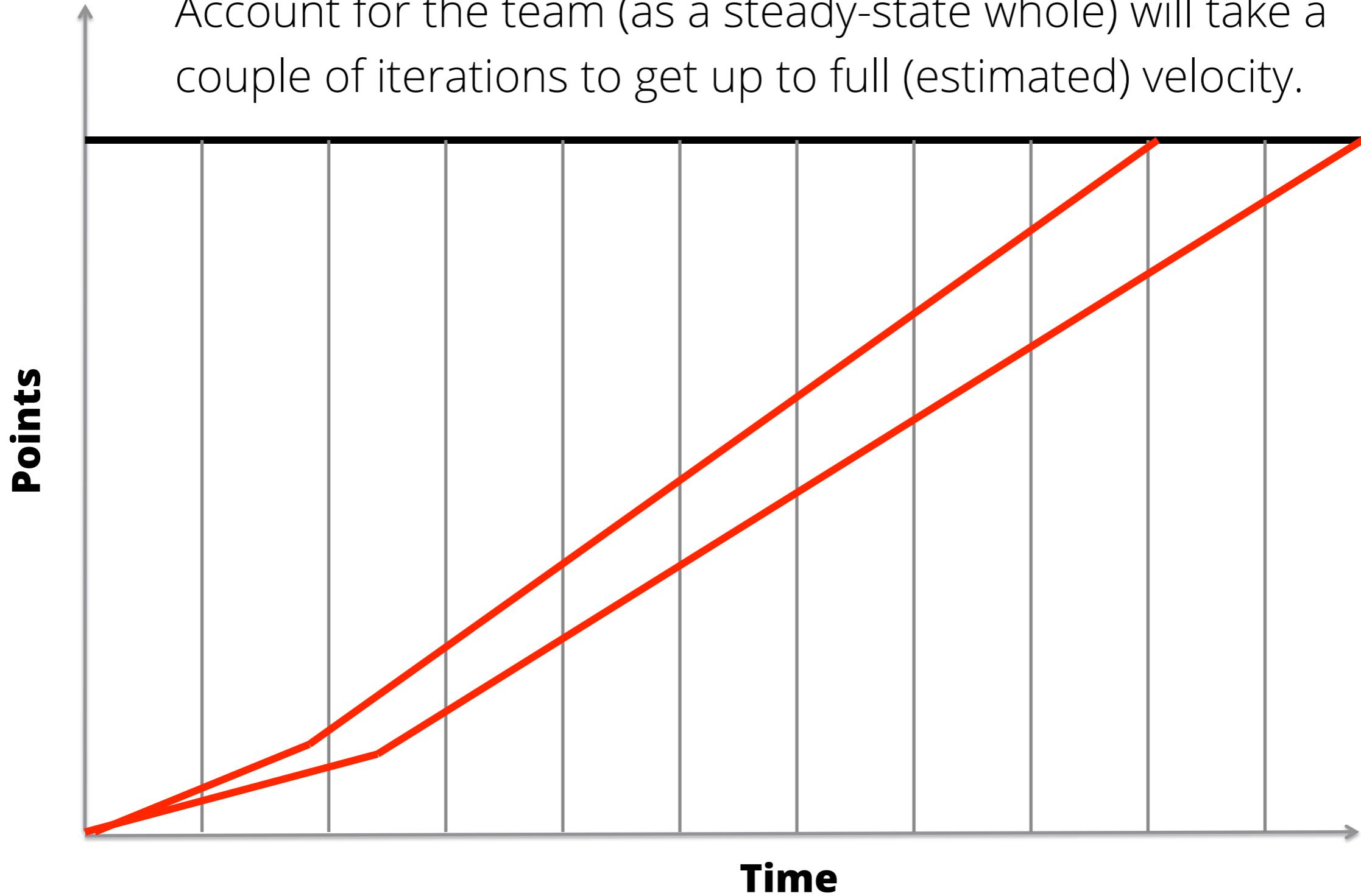
VELOCITY ADJUSTMENT: RISK

Calculate with lower end of the velocity estimate, or to account for risks.



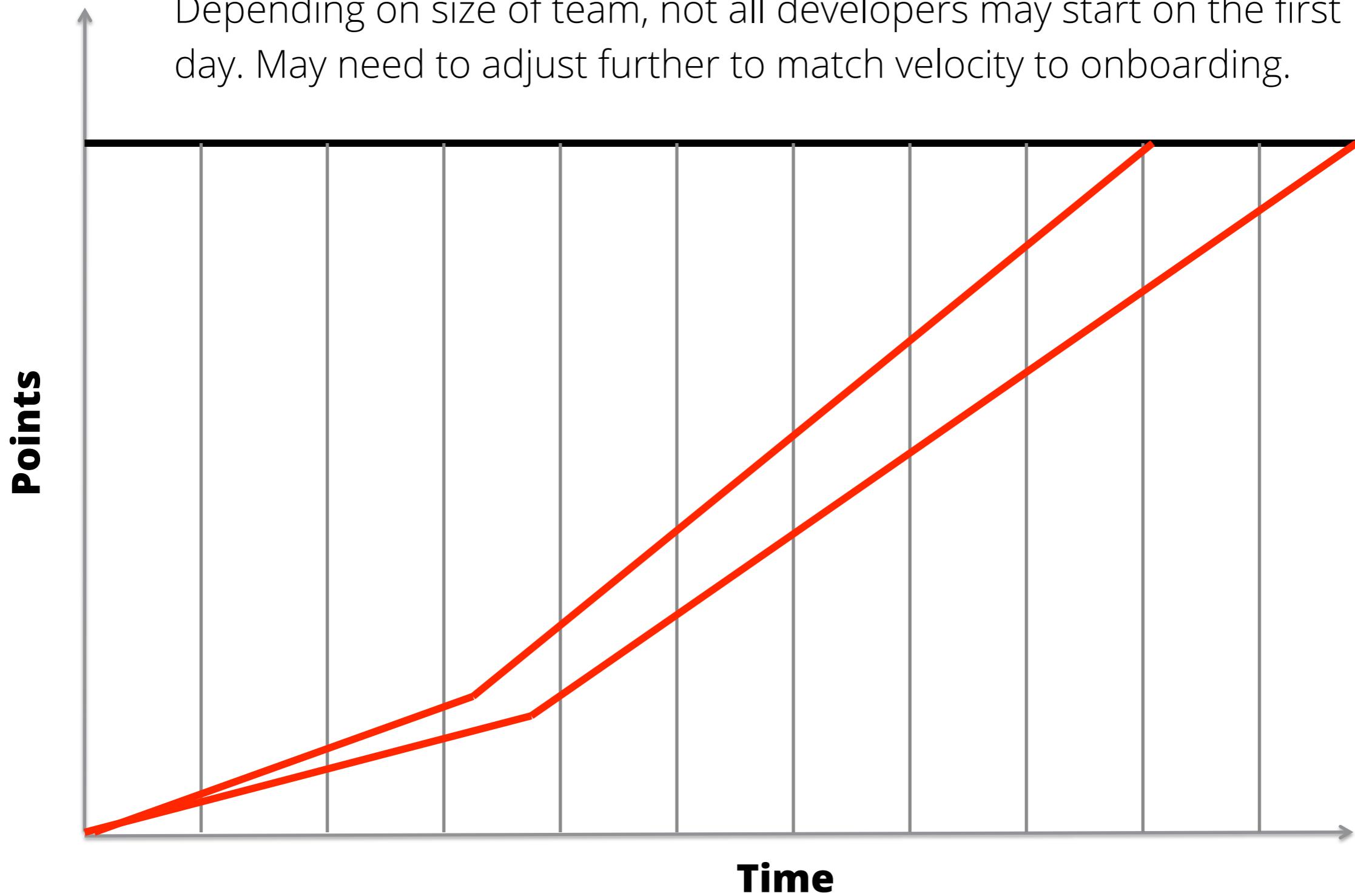
VELOCITY ADJUSTMENT: RAMP UP

Account for the team (as a steady-state whole) will take a couple of iterations to get up to full (estimated) velocity.

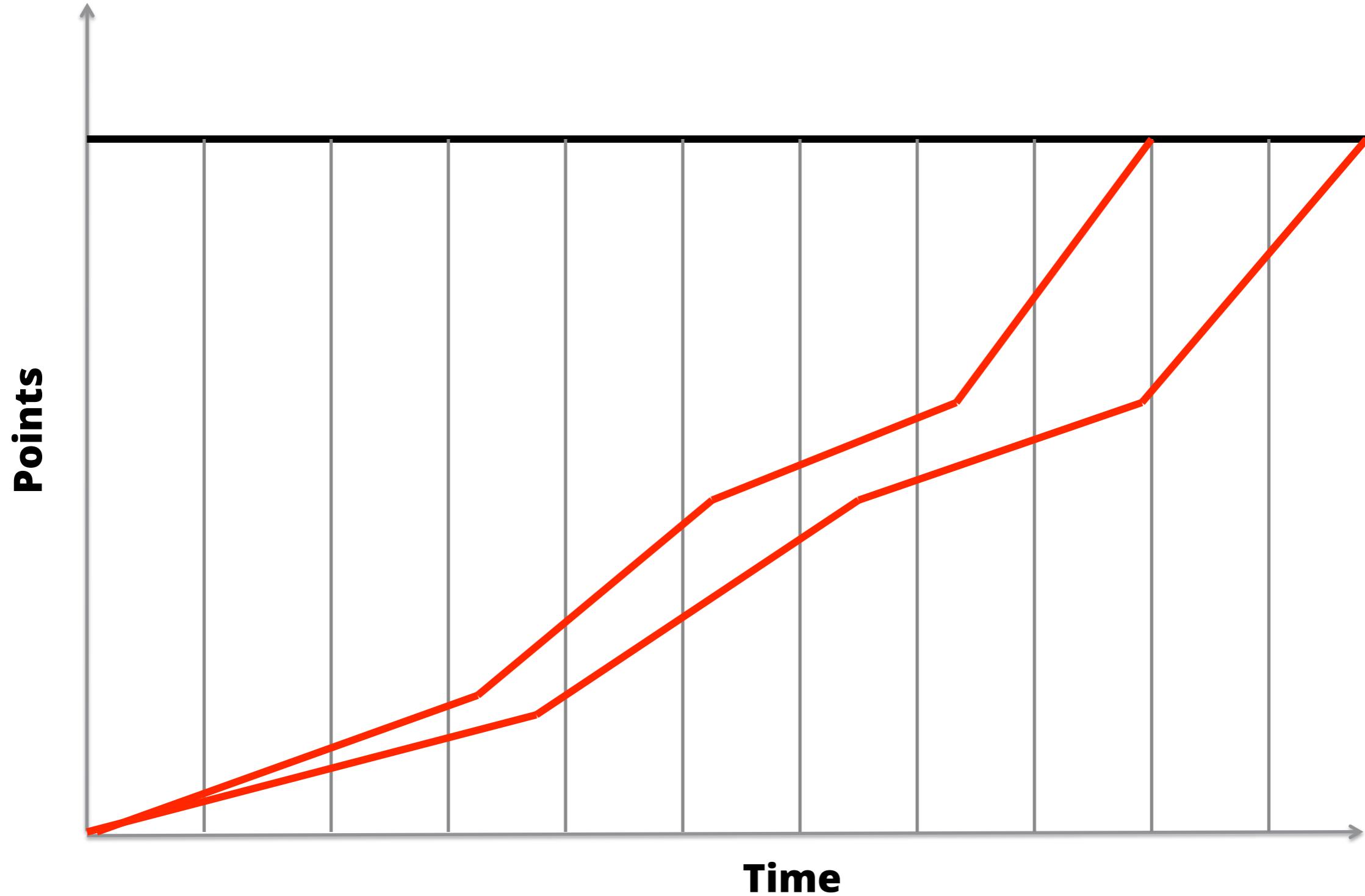


VELOCITY ADJUSTMENT: ONBOARDING

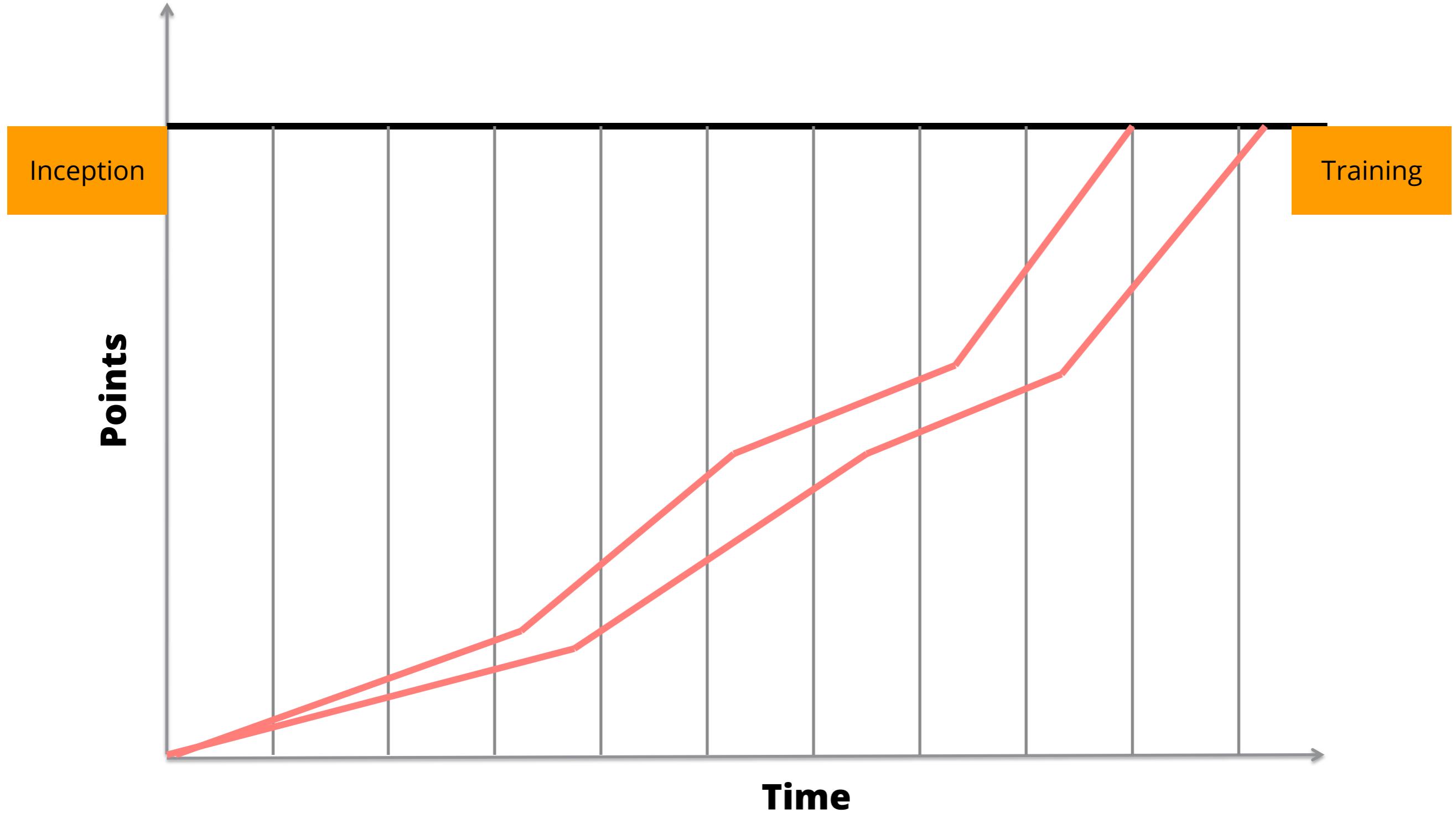
Depending on size of team, not all developers may start on the first day. May need to adjust further to match velocity to onboarding.



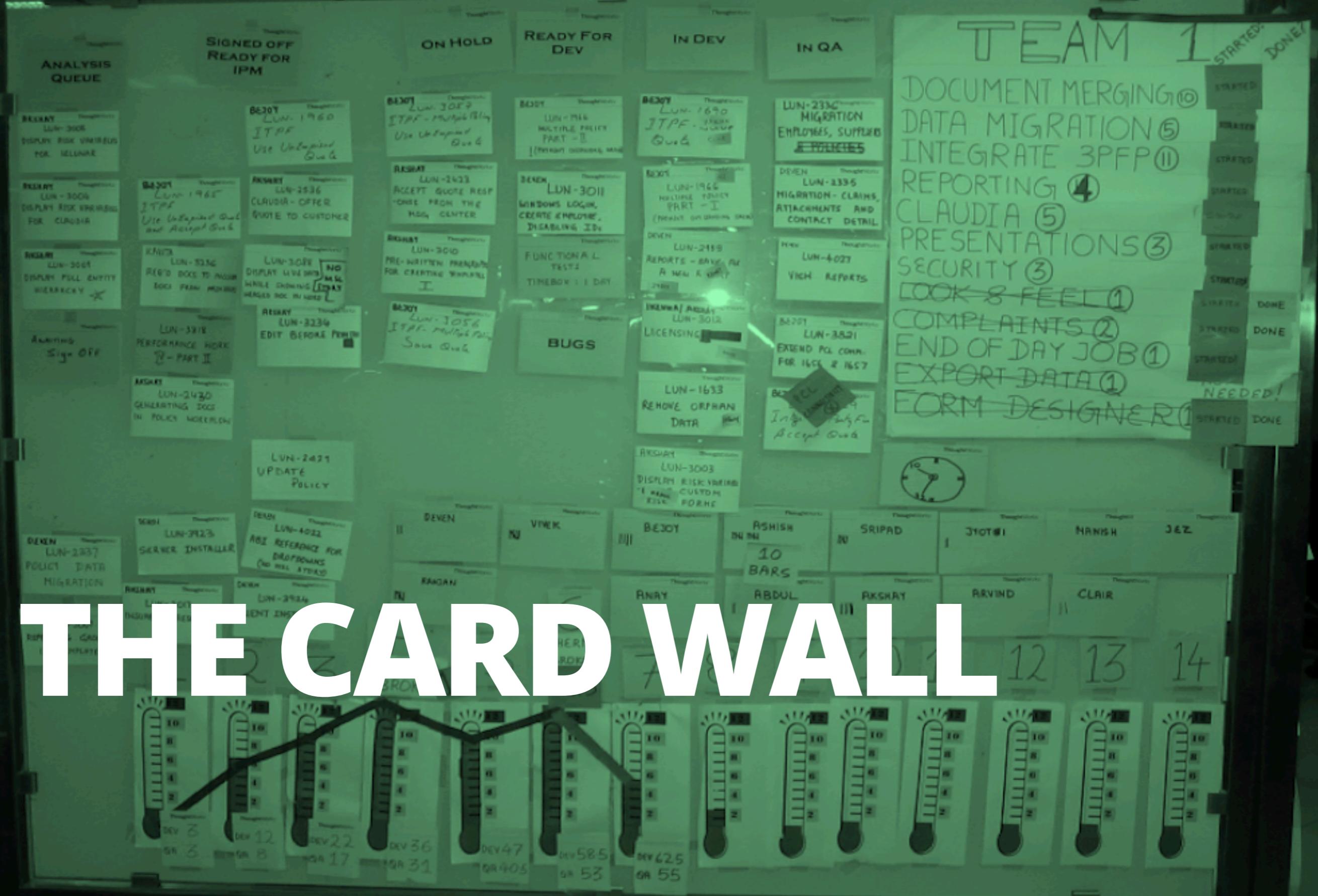
VELOCITY ADJUSTMENT: TIME-OFF



NON-DEVELOPMENT TIME



THE CARD WALL



QUESTIONS?

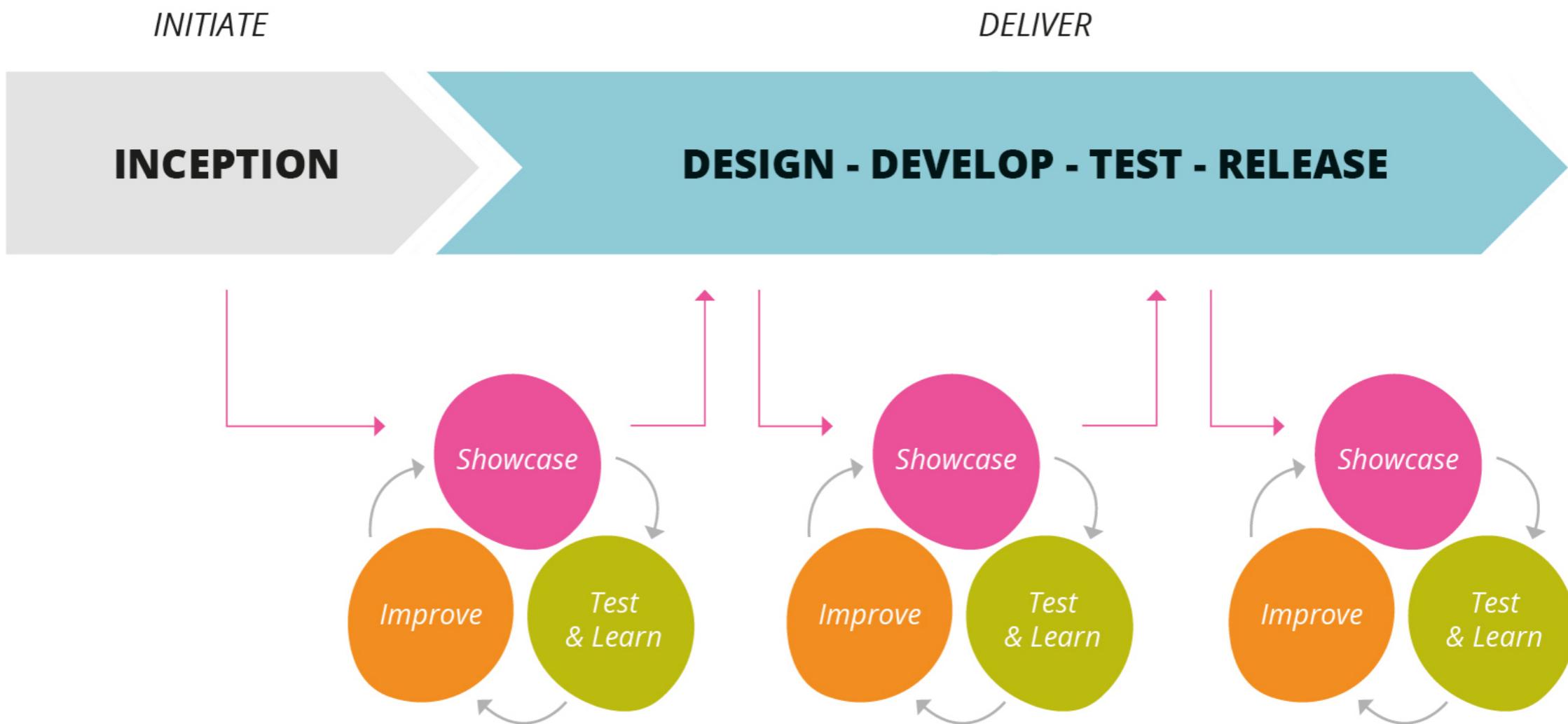
ThoughtWorks®

WRAPPING UP ON THE PROJECT LIFE

IT STARTS WITH THE INCEPTION...



...AND ITERATES THROUGHOUT THE PROJECT



QUESTIONS?

ThoughtWorks®