

AGILE REQUIREMENTS



mavenTHOUGHT

HIGH QUALITY REQUIREMENTS

WHAT'S THE GOAL?

- (good question)

TECHNIQUES

- Verbally
- Use Cases
- Prototyping
- User Stories

COMPLEXITY

- Task Breakdown
- Story points
- Planning poker
- Throw a dice?

PLANNING

- Gantt Chart
- Sprints for releases
- Continuous release
- Story Mapping
- A mix of those

COMMON PROBLEMS

- (Your input here)

CAPTURING REQUIREMENTS

MOVIE LIBRARY SITE

**Post-It
&
Sharpie!**

- Context:
 - Donald likes to watch movies and tell his friends about it. He uses a site that helps him to find movies (old, new, etc), rate them and share with others.
 - He also enjoys the previews and loves to hear when his “favorites” movies are playing in theaters.
- Write a list of interesting things the site could have to provide value for someone like Donald.

WHY USER STORIES?

- Focused on what not how
- Based on what you expect
- Manage scope (light weight)
- Don't include many details
- They can be estimated

WHAT'S THE DIFFERENCE?

- Capture user stories (features) to fill our backlog
- Don't include any details
- Only a vague idea of what it should do
- It should fit in a card
- It's not a contract

PLACEHOLDER FOR CONVERSATION

- Documents are not a great tool for common understanding, so we tell stories
- They spark a conversation (that's the key)
- The story records/reminds of that conversation
- They should be created with the stakeholder/owner

WRITING USER STORIES

FORMAT

- As a **[ROLE1]**, I want to **[ACTION1]**, so I can **[GOAL1]**

Who

What

Why

As a **User**

I want to **register the movies I've seen**

So I can **show them to my friends**

FORMAT II

- In order to **[GOAL]**, as a **[ROLE]**, I want to **[ACTION]**

Why

Who

What

In order to **show the movies I've seen to others**
As a **User**
I want to **register them into the catalog**

INVEST GUIDELINE

- Independent
- Negotiable
- Valuable
- Estimable
- Small
- Testable



■ INVEST - INDEPENDENT

- One user story should be independent of another (as much as possible). Dependencies between stories make planning, prioritization, and estimation much more difficult.
 - As a User, I want to see a dashboard with new movies, my favorites and my friends movies
 - As a User, I want to see new movies
 - As a User, I want to see my favorites
 - As a User, I want to see my friend's movies

INVEST - NEGOTIABLE

- The "Card" of the story is just a short description of the story which do not include details. The details are worked out during the "Conversation" phase.
 - As a User, I want to rate movies comparing director, actors, popularity, social aspect, etc
 - As a User, I want to rate movies
 - As a User, I want to include actors and directors in the reviews

INVEST - VALUABLE

- Each story has to be of value to the customer (either the user or the purchaser). One very good way of making stories valuable is to get the customer to write them.
- As a Developer, I want use a NoSQL db to store movies
- As a developer I want to use Twitter Bootstrap for the UI
- As a User, I want to register movies

INVEST - ESTIMABLE

- The developers need to be able to (relative) estimate (at a ballpark even) a user story to allow prioritization and planning of the story.
 - As a User, I want to have the most secure site
 - As a Reviews, I want to be the only one that sees my private reviews

INVEST - SMALL

- The stories should be implemented in a “small” amount of time, otherwise they are too big and should be broken down.
 - As a User I want to get statistics about movies
 - As a User, I want to see the top 5 searched movies

INVEST - TESTABLE

- Each story should include the *acceptance criteria* that indicates the story is implemented as expected. The owner should participate and define this criteria.
 - The movie site should be easy to use
 - Given I am in the home page
 - When I start typing in the search box
 - Then the auto complete should help me

NON FUNCTIONAL

- How do we write them?
 - Write tests
 - Create a checklist (should be a test)
 - If possible, write it as story

DETAILS

- When's the right time to add details?
- Avoid adding too many details until the story is prioritized to be next
- When doing **Analysis**, then the story gets defined
- Acceptance tests should be included at that point
- If its too big, should be split up

MOVIE LIBRARY SITE II

**Post-It
&
Sharpie!**

- Rewrite your statements as user stories using format I or II
 - As a [ROLE], I want to [ACTION], so I can [GOAL]
 - In order to [GOAL], as a [ROLE], I want to [ACTION]
- Compare with your team
- General Discussion

SCENARIOS

WRITING SCENARIOS

- Each story can be described
- As a sequence of steps
- We call the **alternatives** in the flow **scenarios**

COMMON LANGUAGE

- How do we write scenarios?
- Free form may be too confusing,
- A tight syntax may be too restricting

GHERKIN DSL

- Business readable DSL
- Flush out requirements
- Automated testing
- Used by Cucumber, SpecFlow, jBehave



SCENARIOS

Scenario: List all movies

Given I'm logged in

Step 1

And I have (some data loaded)

Step 2

When I (do some action)

Step 3

Then I (should see expected results)

Step 4

EXAMPLE

Scenario: Write a review for a movie

Given I'm logged in as a Critic

When I write a review for a movie

Then I should see the new review
in the reviews section for the movie

EXAMPLE II

Scenario: See most recent movies first

Given I'm logged in as a User

When I order the list by release date

Then I should see the movies with the release date most recent first

EXCERCISE

Post-It
&
Sharpie!

- Write a story and scenarios
- for **user login** using
- **Given, When, Then**
- **And, But**

GUIDELINES & GOALS

?????
.....

Scenario: Perfect world

Given the application is setup

When I want to use some projects

Then I should be able to load data

And have a great user experience

but no bugs should appear

?????
.....

Scenario: Redirect user to originally requested page
Given a User "dave" exists with password "secret"
And I am not logged in
When I navigate to the home page
Then I am redirected to the login form
When I fill in "Username" with "dave"
And I fill in "Password" with "secret"
And I press "Login"

DESCRIPTIVE STYLE

Scenario: List all projects

Given I'm logged in

And I have some projects stored

When I list the projects

Then I should see all of them

DISCOVER COMPLEXITY

- How many scenarios per feature?
- The more scenarios the more complex
- The more steps the more complex
- If scenarios are unclear, then is time to rethink the feature

DISCOVER FUNCTIONALITY

- When is the right time to write scenarios?
 - During **Inception**?
 - During **Analysis**?
 - During **Development**?
 - During **QA**?

SCENARIO ORDER

- What happens with dependencies?
- How do I use data if I haven't implemented that feature?
- Can devs work independently in different scenarios?


VOTING APP

**Post-It
&
Sharpie!**

- In teams
- Choose top three user stories
- Write scenarios
- Share with the other teams

ESTIMATING USING SCENARIOS

DISCOVER FUNCTIONALITY

- When is the right time to write scenarios?
 - During **Inception**? 
 - During **Analysis**?
 - During **Development**?
 - During **QA**?

ORDER MATTERS

- Can every feature be independent?

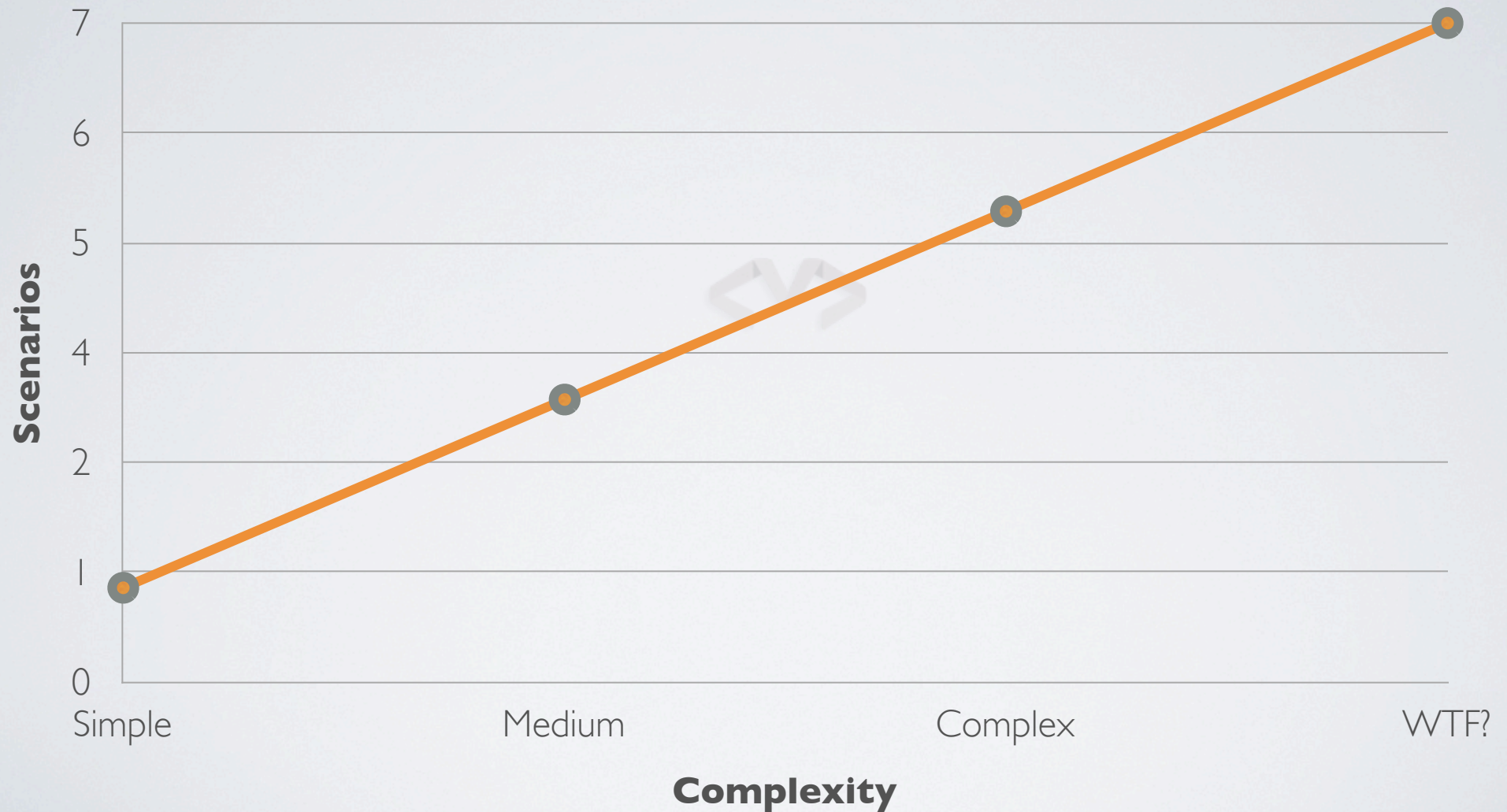


SIZE MATTERS

- Not every story is the same size



COMPLEXITY \propto AMOUNT



COMPLEXITY \propto CLARITY

- Too many steps
- Takes more than 5 minutes to write
- You read it the next day and have no clue what it means

RECIPE - STEP #1

Make every story around the **same size**



- One or two scenarios most
- If more, break it down to more stories

RECIPE - STEP #2

Put a **time limit** (days) to each feature



- For example: A story has to be completed in 1 or two days at most

RECIPE - STEP #3

Choose the **simplest** story to flush out requirements and implementation

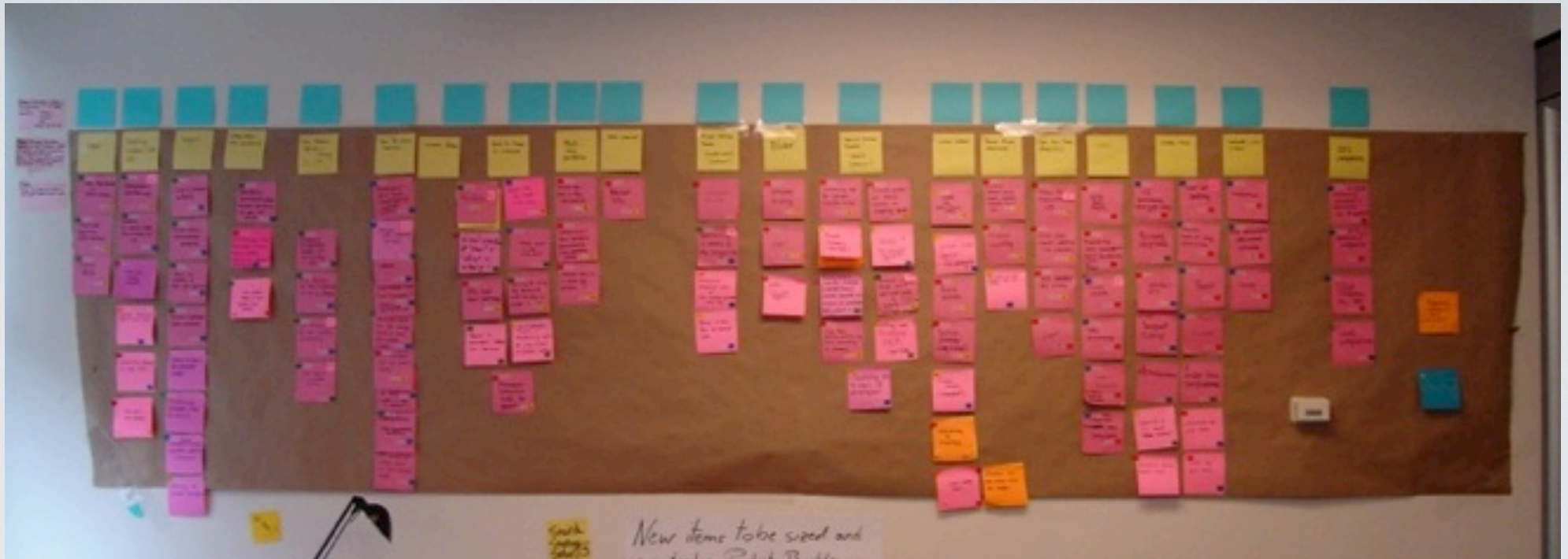
- For example: User & password login, OAuth will come later

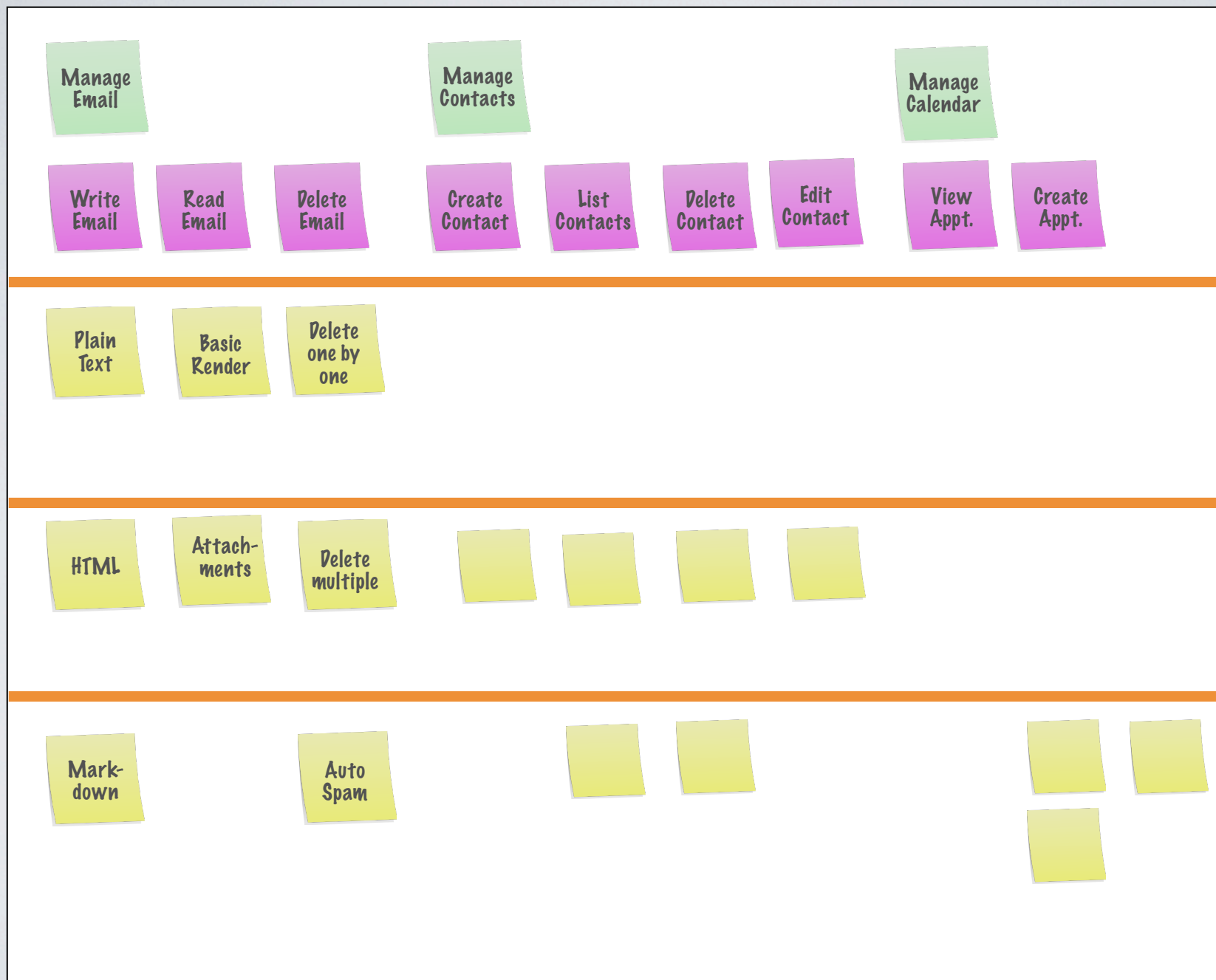
USER STORY MAPPING

**User Story Mapping is
an approach to
Organizing and
Prioritizing user stories**

STORY MAP VS BACKLOG

- Clearly identifies value
- Track finished stories
- Helps prioritize stories
- Separate stories in releases easily





Release 1

Release 2

Release 3

User Activities

User Tasks

User Stories

User Stories

User Stories

Release 1

Release 2

Release 3

Priority

Time



Release 1

Release 2

Release 3

The first release
represents
the minimum viable
release

BUDGET
TIME
SCOPE

Fixed

Fixed

?????

Better to have
80% 100% done
Than
100% 80% done

NARROWING THE GAP

ACCEPTANCE CRITERIA

- Write the expected criteria before the implementation starts
- Developers will implement the feature until the criteria is satisfied
- QA will validate against the same criteria

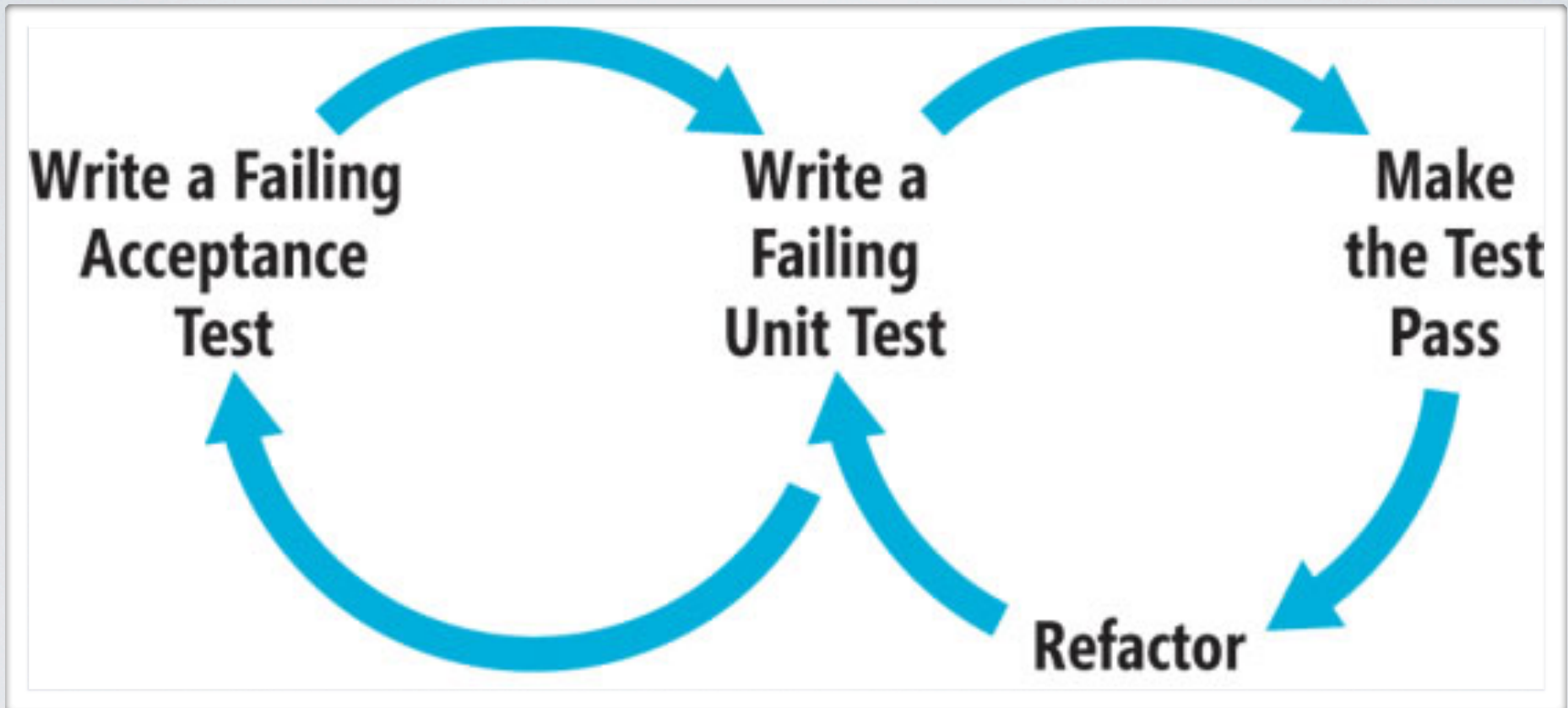
BENEFITS

- Discover the feature
- Testing all the way
- Traceability
- Quality every step of the process

ROLES

- Who writes the feature?
- Who implements the feature?
- Who validates the feature?
- What's the role of QA, PM, etc?

OUTSIDE IN APPROACH



RUNNABLE FEATURES

- Features describe functionality
- What if we could run them?
- Then features would validate functionality
- Becoming live documentation

TOOLS GALORE

- Lots of tools available that use Gherkin:
 - Cucumber / rSpec
 - jBehave
 - Specflow
 - Scalatest
 - etc

WHERE SHOULD THEY GO?

- Acceptance tests?
- Integration tests?
- Unit tests?

ACCEPTANCE TEST

- Black box testing
- Crossing all layers
- Should cover all scenarios
- External subsystems may be mocked

INTEGRATION TEST

- More than one class
- Still some parts can be mocked
- Partial functionality of subsystem

UNIT TEST

- Test for a class or method
- No external dependencies
- Small
- Clear

TDD

- First write a test that fails (**RED**)
- Write code to make it pass (**GREEN**)
- Check if code can be improved (**REFACTOR**)
- Start again until it's done

WHEN TDD IS NOT ENOUGH

- Legacy Code
- Refactoring is not viable
- Verify functionality across layers
- Validate feature end to end

DEMO

SUMMARY

Benefits

Challenges

What's next?

BENEFITS

- Easier planning
- Discovers functionality and complexity
- No particular skill required
- Easy Adoption
- Puts the whole team on the same page
- Simplifies QA process
- Narrows the gap between expectations and actual implementation

CHALLENGES

- Different approach
- The roles get blurred
- Team effort
- Others?

WHAT'S NEXT

- Take one idea/technique that you think could be beneficial
- Use the next four weeks to implement it
- Register challenges, issues and results
- Share with your peers (and with me if possible)

QUESTIONS?

RESOURCES

- Contact me: amir@barylko.com, @abarylko
- Download: <http://www.orthocoders.com/presentations>