SCRUM

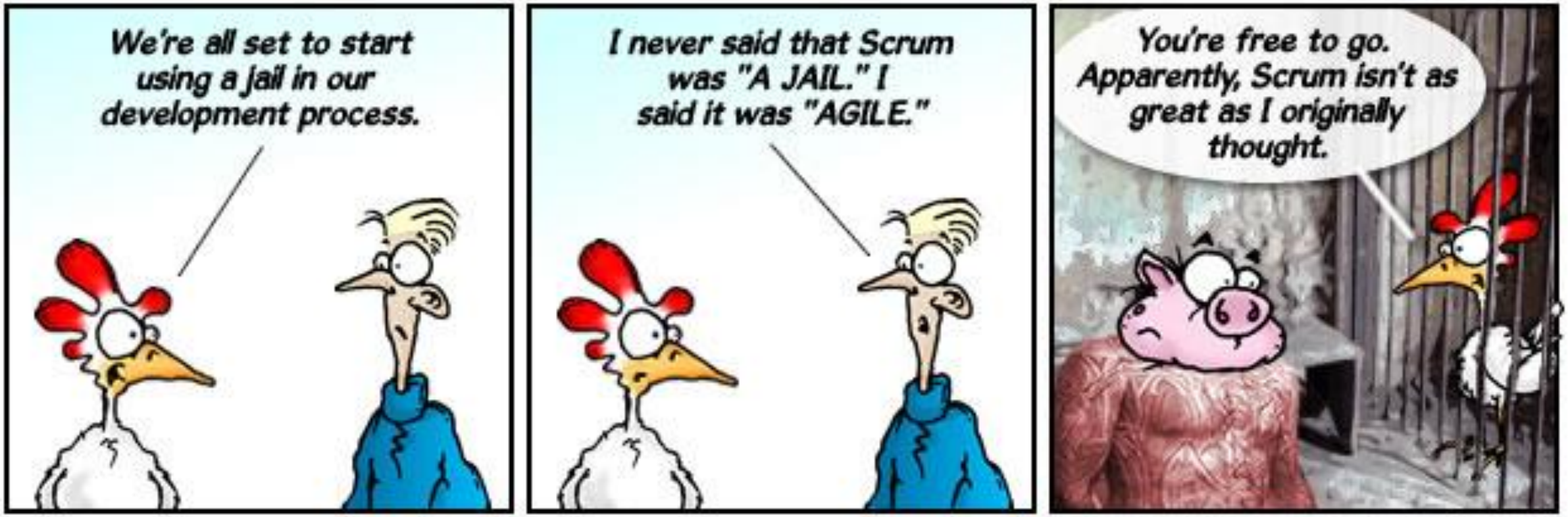
EPITA Information Management Master

Scrum Agile

Module 2

Olivier BERTHET

SCRUM


We're all set to start using a jail in our development process.
I never said that Scrum was "A JAIL." I said it was "AGILE."
You're free to go. Apparently, Scrum isn't as great as I originally thought.
By Clark & Vizdos
© 2006 implementingscrum.com

Agenda

- **Session 1** **15th of March 2018**
  - Definition , history
  - Agile principles, comparison waterfall versus scrum, Scrum benefits
  - Scrum framework and main principles
  - Scrum artifacts , product backlog

- **Session 2** **16th of March 2018**
  - User stories
  - Planning : scrum planning principles , product and release planning

Agenda

- **Session 3** **17th of March 2018 at 9 AM**
  - **Estimating and velocity**
  - **Poker game**

- **Session 4** **30th of March 2018**
  - **Sprinting : sprint planning , sprint execution, sprint review and retrospective**

Agenda

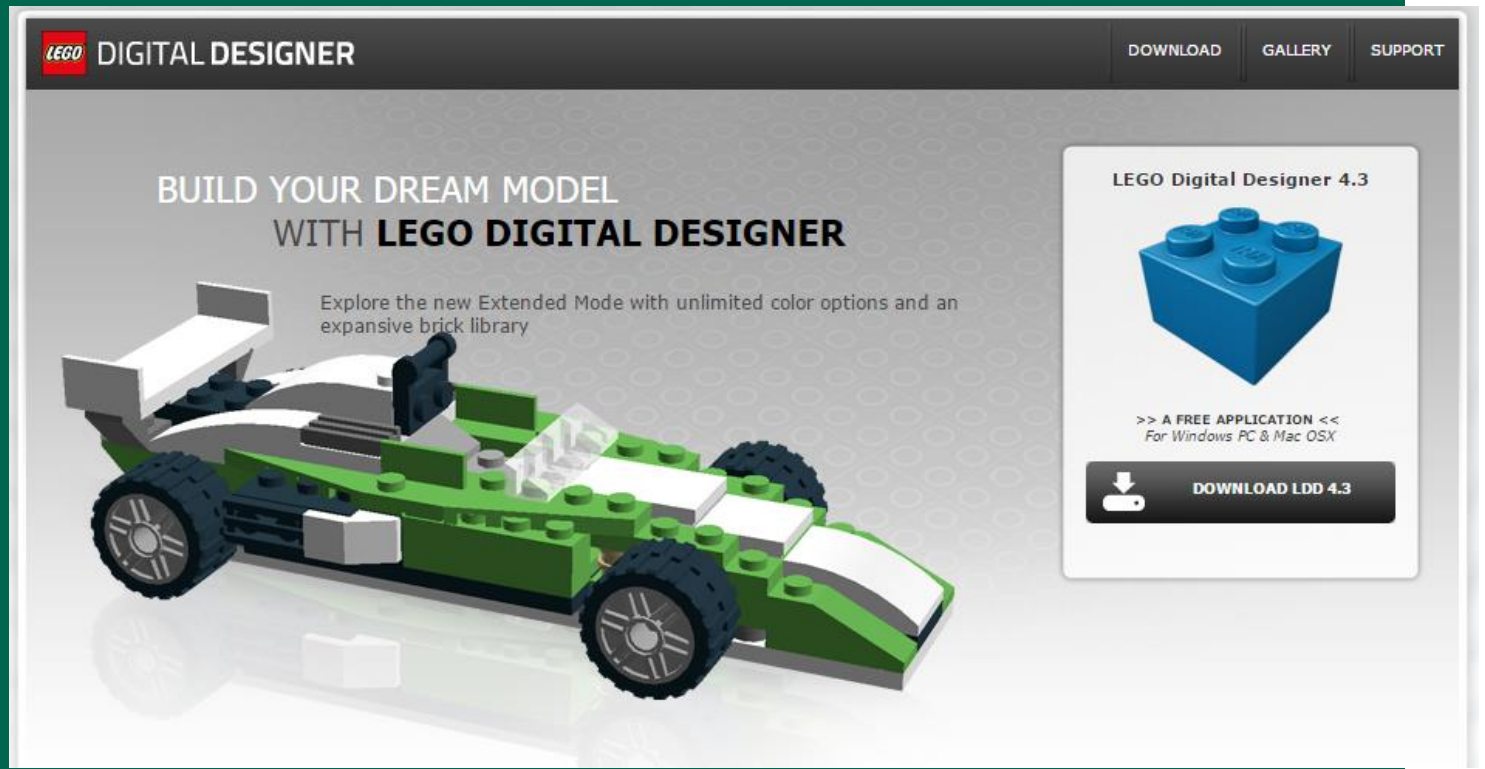
- **Session 5** **20th of April 2018**
  - **Exam preparation : sprint planning**

- **Session 6** **20th of April 2018**
  - **Exam : Execution of sprints**

SCRUM

Lego Digital Designer

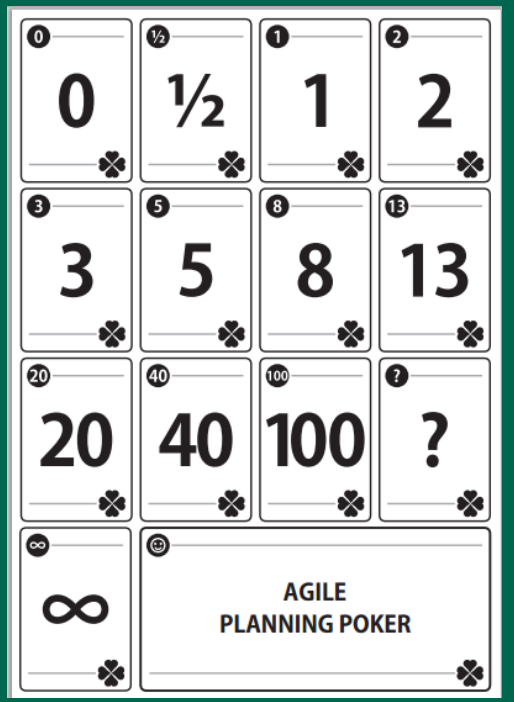
- http://ldd.lego.com/en-us/



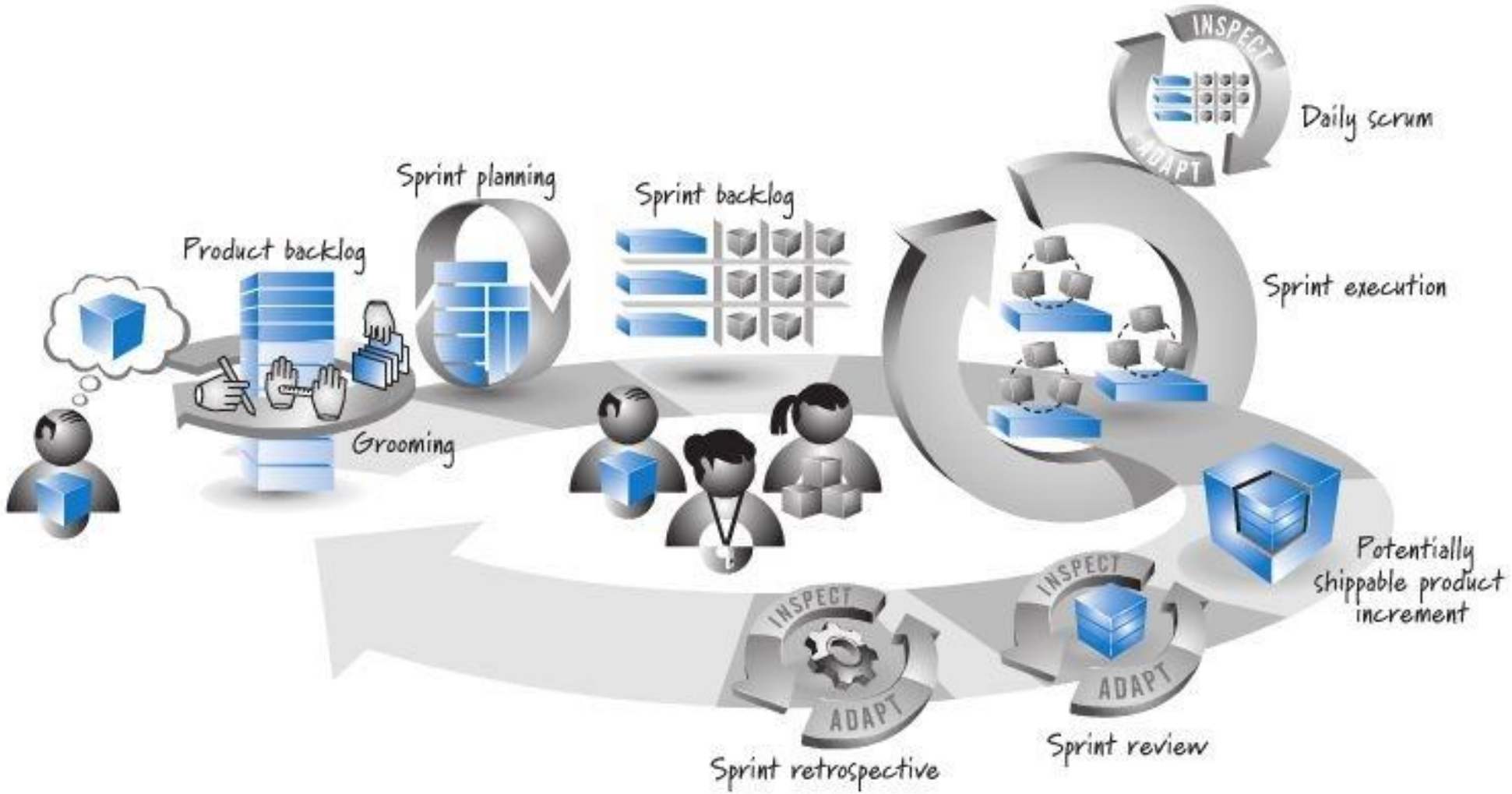
SCRUM

Planning poker

- http://wwwis.win.tue.nl/2R690/doc/agile_planning_poker.pdf



SCRUM

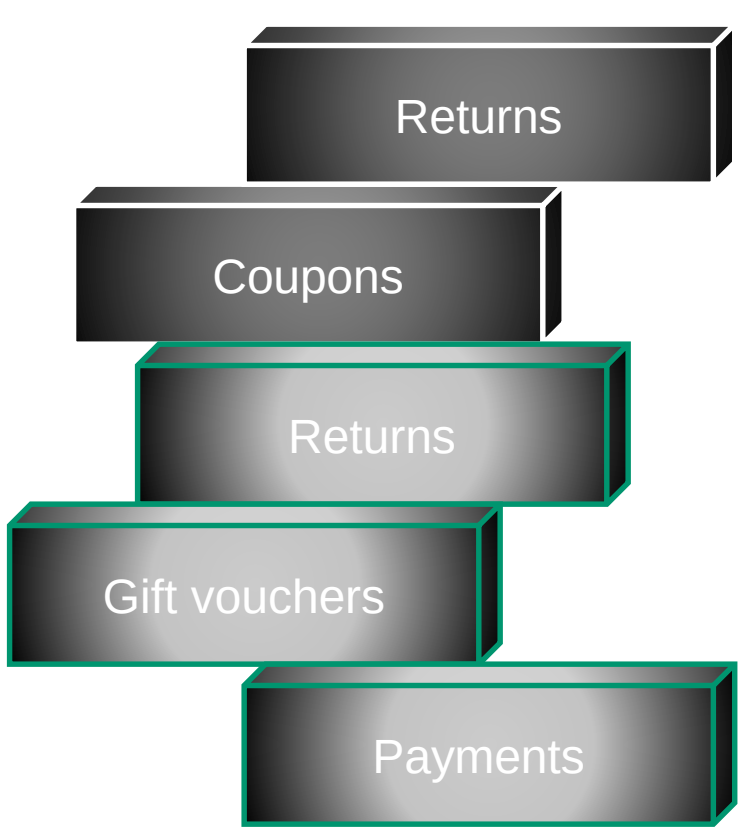

INSPECT
ADAPT
Daily scrum
Sprint planning
Sprint backlog
Product backlog
Sprint execution
Grooming
Potentially shippable product increment
INSPECT
ADAPT
Sprint retrospective
INSPECT
ADAPT
Sprint review

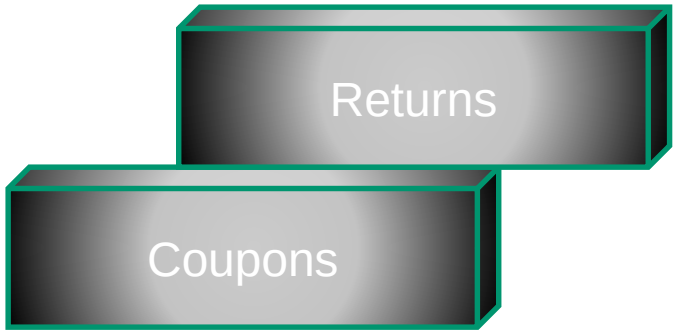
SCRUM

SCRUM


EXTREME PROGRAMMING
I CAN'T GIVE YOU ALL OF THESE FEATURES IN THE FIRST VERSION.
www.dilbert.com scottadams@aol.com
AND EACH FEATURE NEEDS TO HAVE WHAT WE CALL A "USER STORY."
© 2002 United Feature Syndicate, Inc.
OKAY, HERE'S A STORY: YOU GIVE ME ALL OF MY FEATURES OR I'LL RUIN YOUR LIFE.

SCRUM





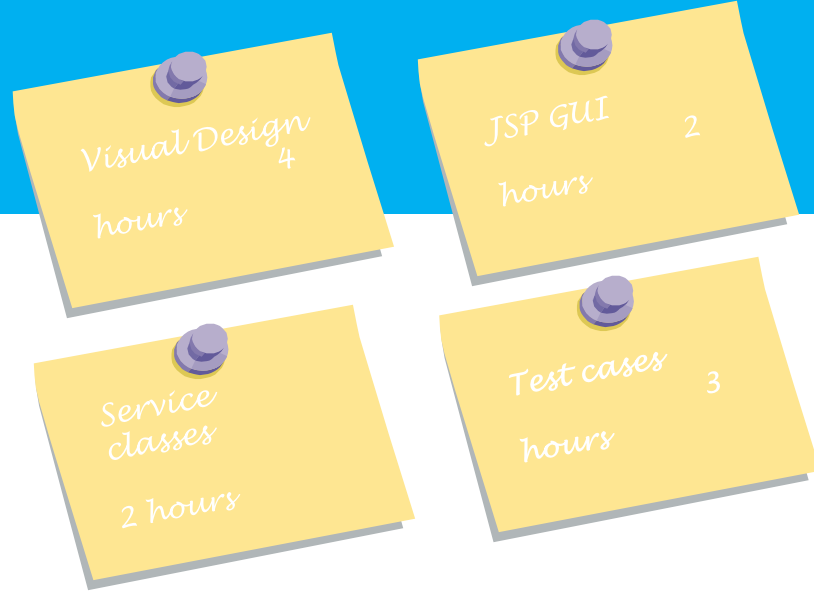
Prioritized Product backlog

Sprint backlog

Sprint goal

EPITA
ÉCOLE D'INGÉNIEURS EN INFORMATIQUE

SCRUM



Returns

Sprint backlog

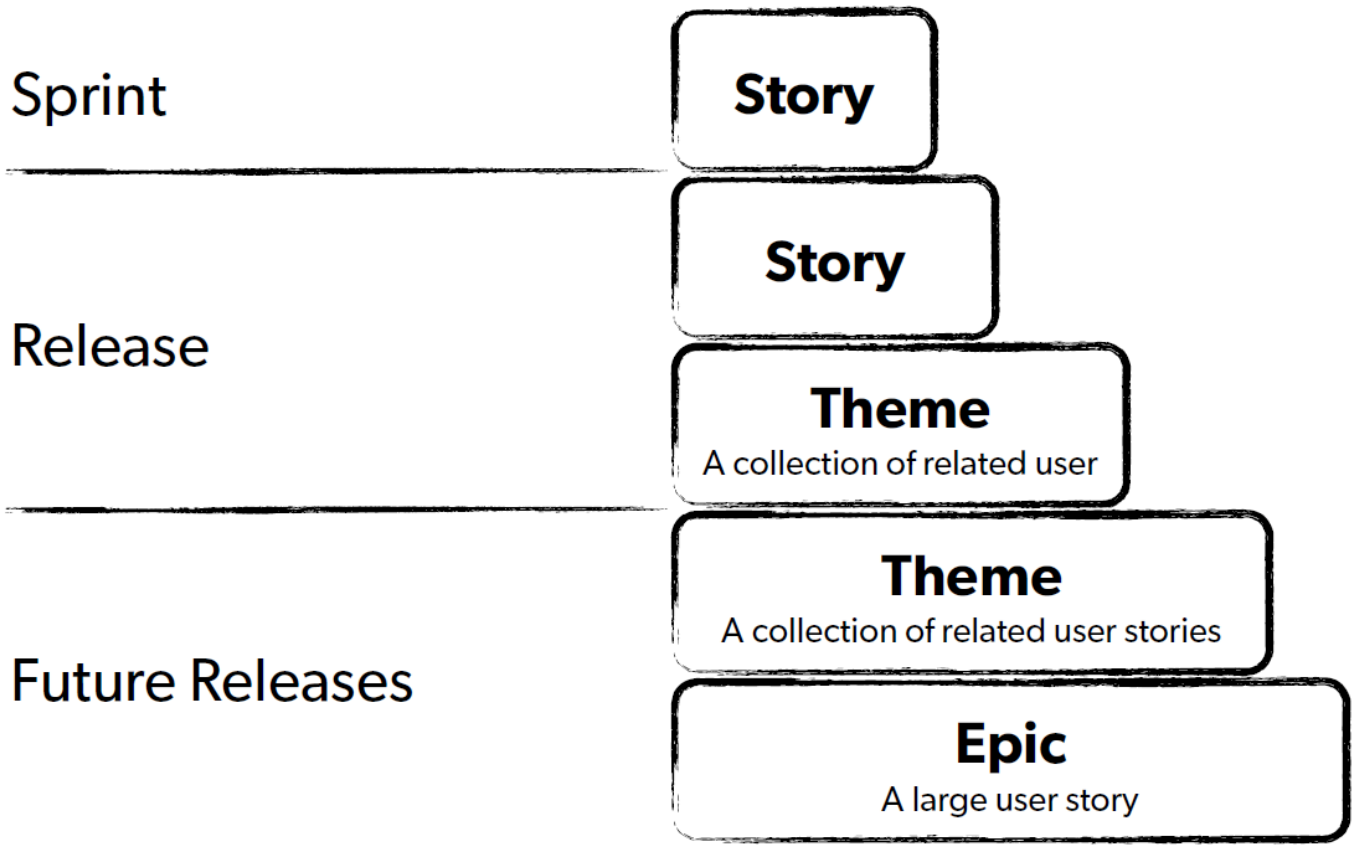
Team builds

Tasks

Coupons

SCRUM

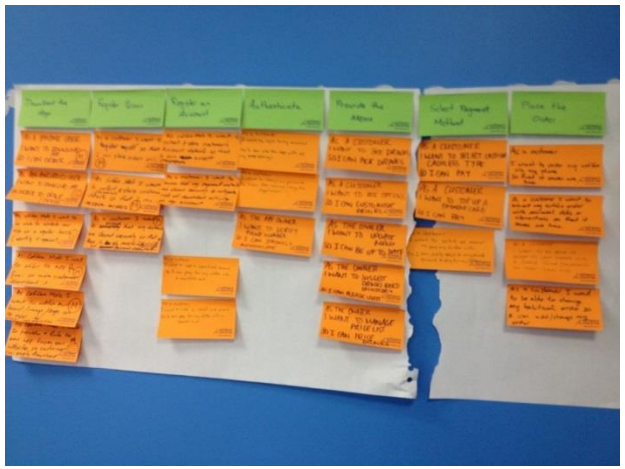
The Backlog Iceberg



SCRUM

What is a user story ?

- **Wikipedia defines it as *“a software system requirement formulated as one or more sentences in the everyday or business language of the user”***
- **A concise, written description of a piece of functionality that will be valuable to a user (or owner) of the software.**



Key Principles for Agile Requirements

- **Active user involvement is imperative**
- **Agile teams must be empowered to make decisions**
- **Requirements emerge and evolve as software is developed**
- **Agile requirements are ‘barely sufficient’**
- **Requirements are developed in small, bite-sized pieces**
- **Enough’s enough – apply the 80/20 rule**
- **Cooperation, collaboration and communication between all team members is essential**

SCRUM

Layman’s definition

- ***“it is a quicker way of detailing requirements without having to create large documents and minimizing the time to administer the documents through the entire life-cycle of the project”.***

Requirements are a Communication Problem

- **Written requirements**
  - can be well thought through, reviewed and edited
  - provide a permanent record
  - are more easily shared with groups of people
  - time consuming to produce
  - may be less relevant or superseded over time
  - can be easily misinterpreted

Requirements are a Communication Problem

- **Verbal requirements**
  - instantaneous feedback and clarification
  - information-packed exchange
  - easier to clarify and gain common understanding
  - more easily adapted to any new information known at the time
  - can spark ideas about problems and opportunities

Gathering User Stories

- **User Interviews**
  - Select right interviewees
  - Ask open-ended, context-free questions
- **Questionnaires**
  - Larger population of users
  - When you need specific answers to questions
- **Observation**
  - Best for in-house developments
- **Story writing workshops**

SCRUM

A picture is worth a thousand words

User Stories

**seek to combine the strengths
of written and verbal communication,
where possible supported by a picture.**

User Story Cards have 3 Cs

- **Card - A written description of the user story for planning purposes and as a reminder**
- **Conversation - A section for capturing further information about the user story and details of any conversations**
- **Confirmation - A section to convey what tests will be carried out to confirm the user story is complete and working as expected**

User Story Description

As a [user role] I want to [goal]
so I can [reason]

For example:

- As a registered user I want to log in
  so I can access subscriber-only content

User Story Description

- **Why (reason)**
  - gives clarity as to why a feature is useful
  - can influence how a feature should function
  - can give you ideas for other useful features that support the user's goals

SCRUM

User Story Example: Front of Card

User Story Example: Back of Card

Confirmation

1. Success – valid user logged in and referred to home page.
   a. 'Remember me' ticked – store cookie / automatic login next time.
   b. 'Remember me' not ticked – force login next time.

2. Failure – display message:
   a) "Email address in wrong format"
   b) "Unrecognised user name, please try again"
   c) "Incorrect password, please try again"
   d) "Service unavailable, please try again"
   e) Account has expired – refer to account renewal sales page.

SCRUM

How detailed should a User Story be?

Detailed enough for the team to start work from, and further details to be established and clarified at the time of development.

INVEST in Good User Stories

- **Independent**
  - User Stories should be as independent as possible
- **Negotiable**
  - User Stories are not a contract. They are not detailed specifications. They are reminders of features for the team to discuss and collaborate to clarify the details near the time of development.
- **Valuable**
  - User Stories should be valuable to the user (or owner) of the solution. They should be written in user language. They should be features, not tasks.
- **Estimatable**
  - User Stories need to be possible to estimate. They need to provide enough information to estimate, without being too detailed
- **Small**
  - User Stories should be small. Not too small. But not too big
- **Testable**
  - User Stories need to be worded in a way that is testable, i.e. not too subjective and to provide clear details of how the User Story will be tested.

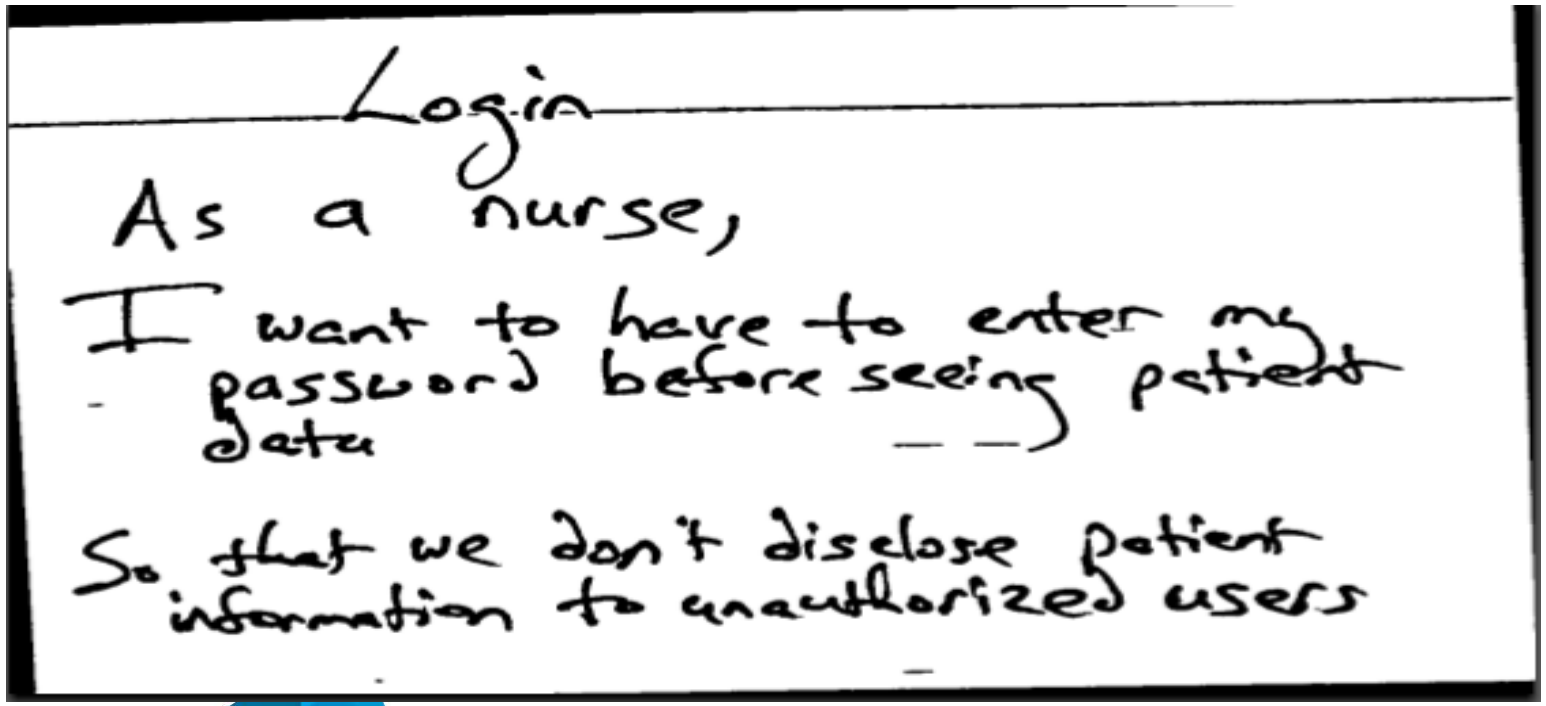
SCRUM

User Story Aspects

- **As a nurse, I want to have to enter my password before seeing patient data, so that we don't disclose patient information to unauthorized users.**

- **As a <role>**
  I want <feature>
  So that <business value>

- **Card**
- **Conversation**
- **Confirmation**



SCRUM

Examples of user stories

- **Sales Manager**
- **See the number of orders completed per month**
- **Track online sales numbers**

As a Sales Manager,

I want to see the number of orders completed per month,

So that I can track online sales numbers

SCRUM

Examples of user stories

- **Customer**
- **Search for a book by title**
- **Find a book quickly online**

As a Customer,

I want to search for a book by title,

So that I can find a book quickly online

SCRUM

Examples of user stories

- **Customer**
- **Store my credit card online**
- **Make multiple purchases quickly**

As a Customer,

I want to store my credit card online,

So that I can make multiple purchases quickly

Examples of user stories

- **Customer**
- **Delete a book from my shopping cart**
- **Remove unwanted items from my cart**

As a Customer,

I want to delete a book from my Shopping Cart

So that I can remove unwanted items from my cart

SCRUM

Examples of user stories

- **Customer**
- **Fill in a suggestion from**
- **Leave feedback for the vendors**

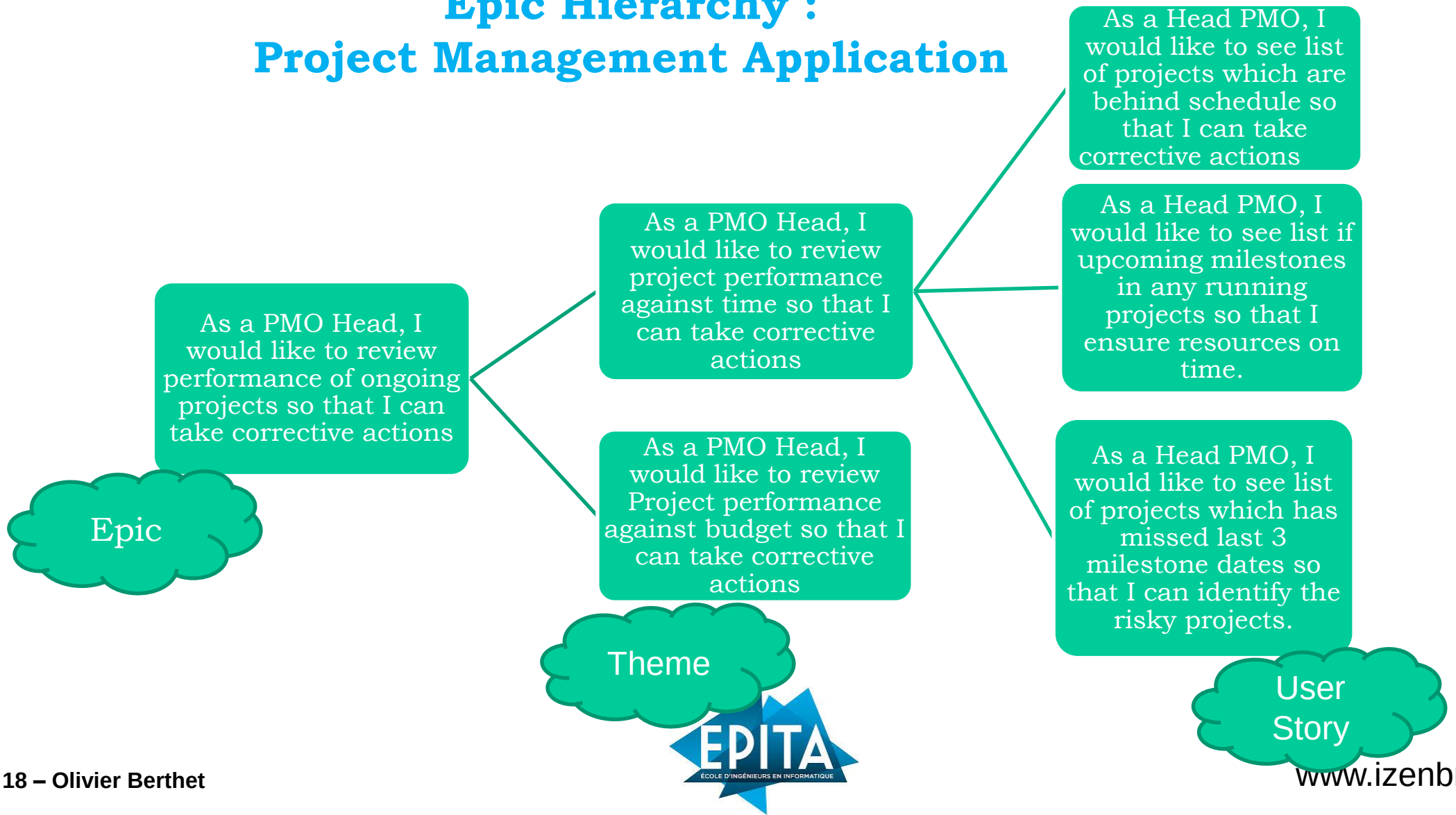
As a Customer,

I want to fill in a Suggestion Form

So that I can leave feedback for the vendors

SCRUM

Epic Hierarchy :
Project Management Application



Definition of Done

- **Definition of Done must describe exactly what “done” means**
  - **Careful attention Must be payed when defining the DoD**
  - **The scrum team must challenge the DoD, if necessary**
  - **“What’s not in DoD, is not needed”**
  - **Item is either “done” or “not done”**

Definition of Done

- **Example:**
  - **Story: Picture upload**
    - **end user can upload his/her picture from profile settings page**
    - **picture is shown on the left upper corner of the profile page**
    - **picture is scaled to fit the profile picture box on the profile page**
    - **functional tests are passed**
    - **regression tests are passed**
    - **design documents are updated**
    - **user's guide is updated**
- **Does not define any details of the implementation!**

Acceptance criteria

- **Verifiable and testable criteria that can be tested based on THEN clause**
- **These are essentially tests – Conditions of satisfaction**
- **Example:**
  - **As a user, I can cancel a reservation.**
  - **Verify that a premium member can cancel**
  - **Verify that a email confirmation is sent**
  - **Verify that the hotel is notified of any cancelation**

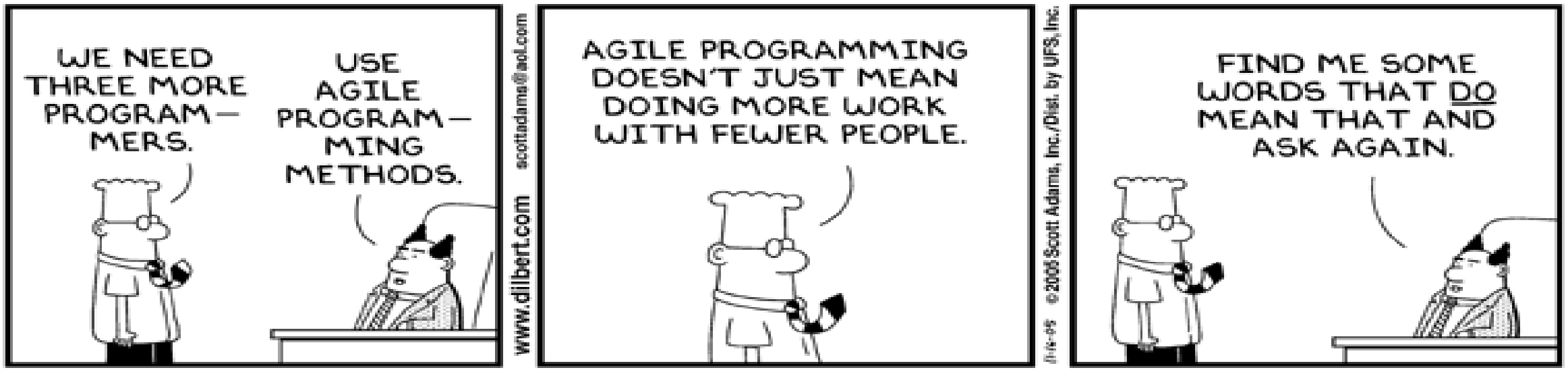
  - **These acceptance criteria can become developer tasks**

User Stories Summary

- **User Stories combine written and verbal communications, supported with a picture where possible**
- **User Stories should describe features that are of value to the user, written in a user's language**
- **User Stories detail just enough information and no more**
- **Details are deferred and captured through collaboration just in time for development**
- **Test cases should be written before development, when the User Story is written**
- **User Stories should be Independent, Negotiable, Valuable, Estimatable, Small and Testable**

SCRUM

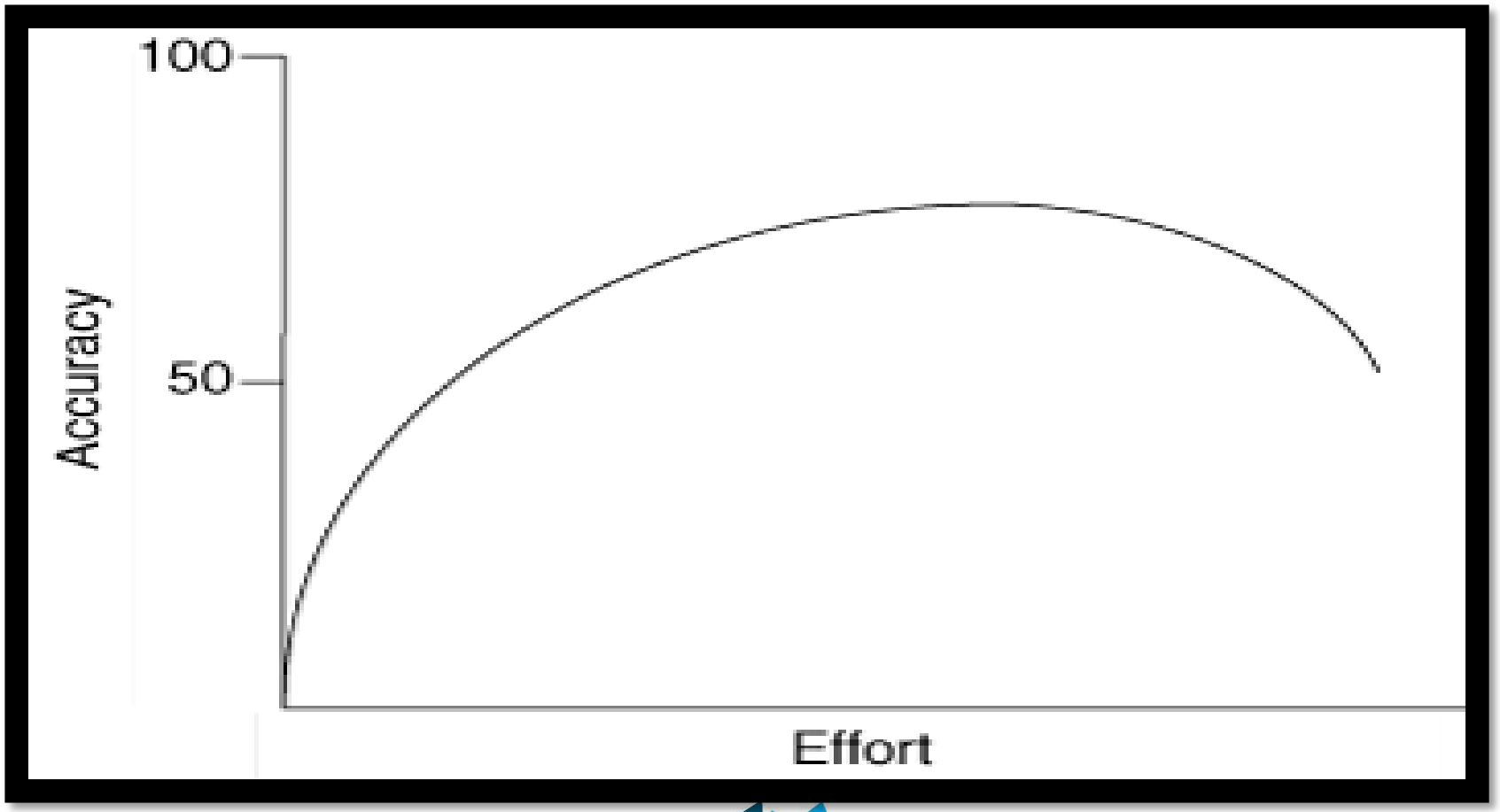
Dilbert's World



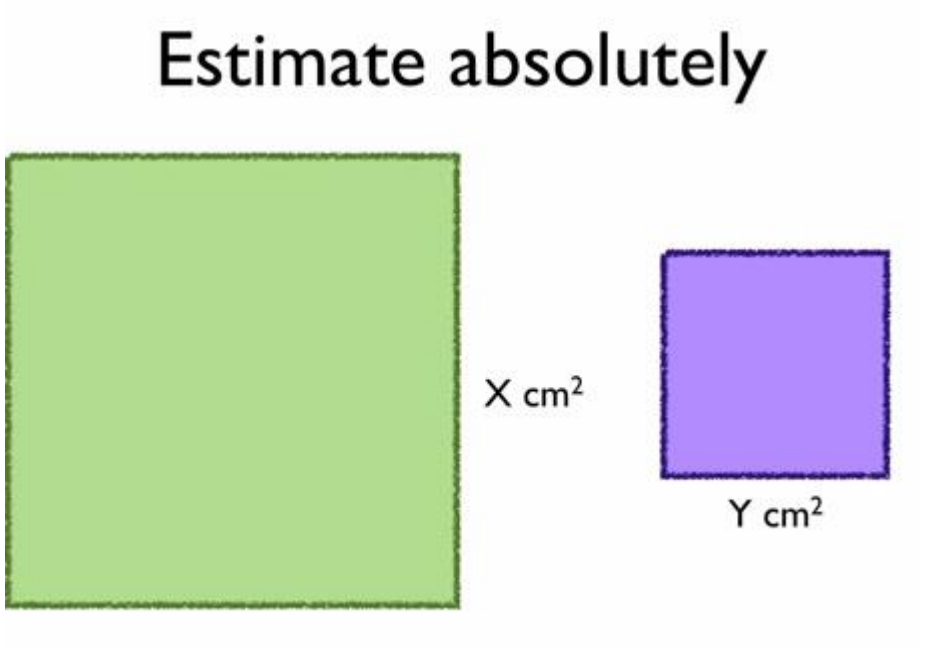
Estimation

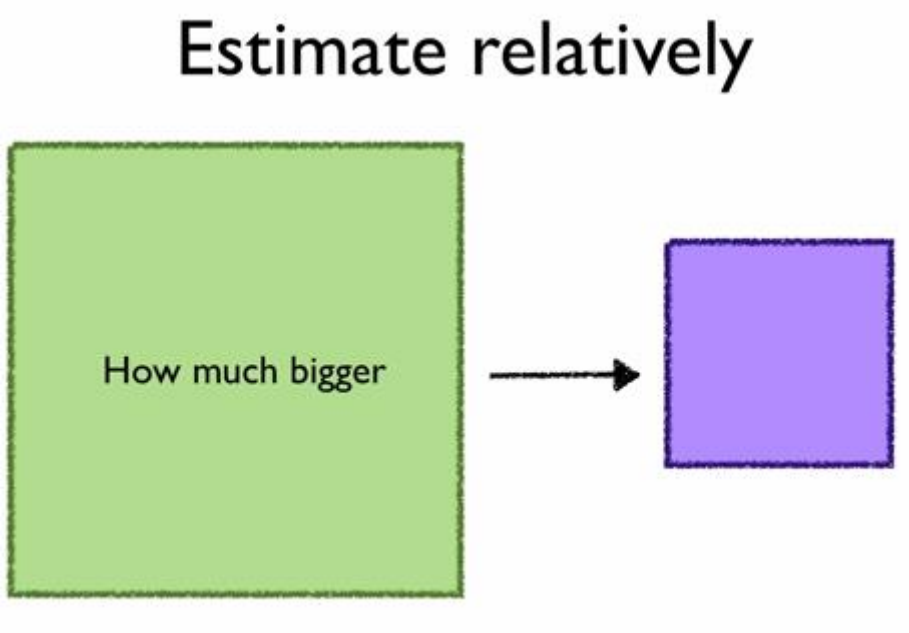
- **Estimation is one of the harder parts of a software project**
- **Nearly 2/3 of projects significantly overrun their cost estimates**
- **64% of the features included in products are rarely or never used**
- **The average project exceeds its schedule by 100%**

Estimation

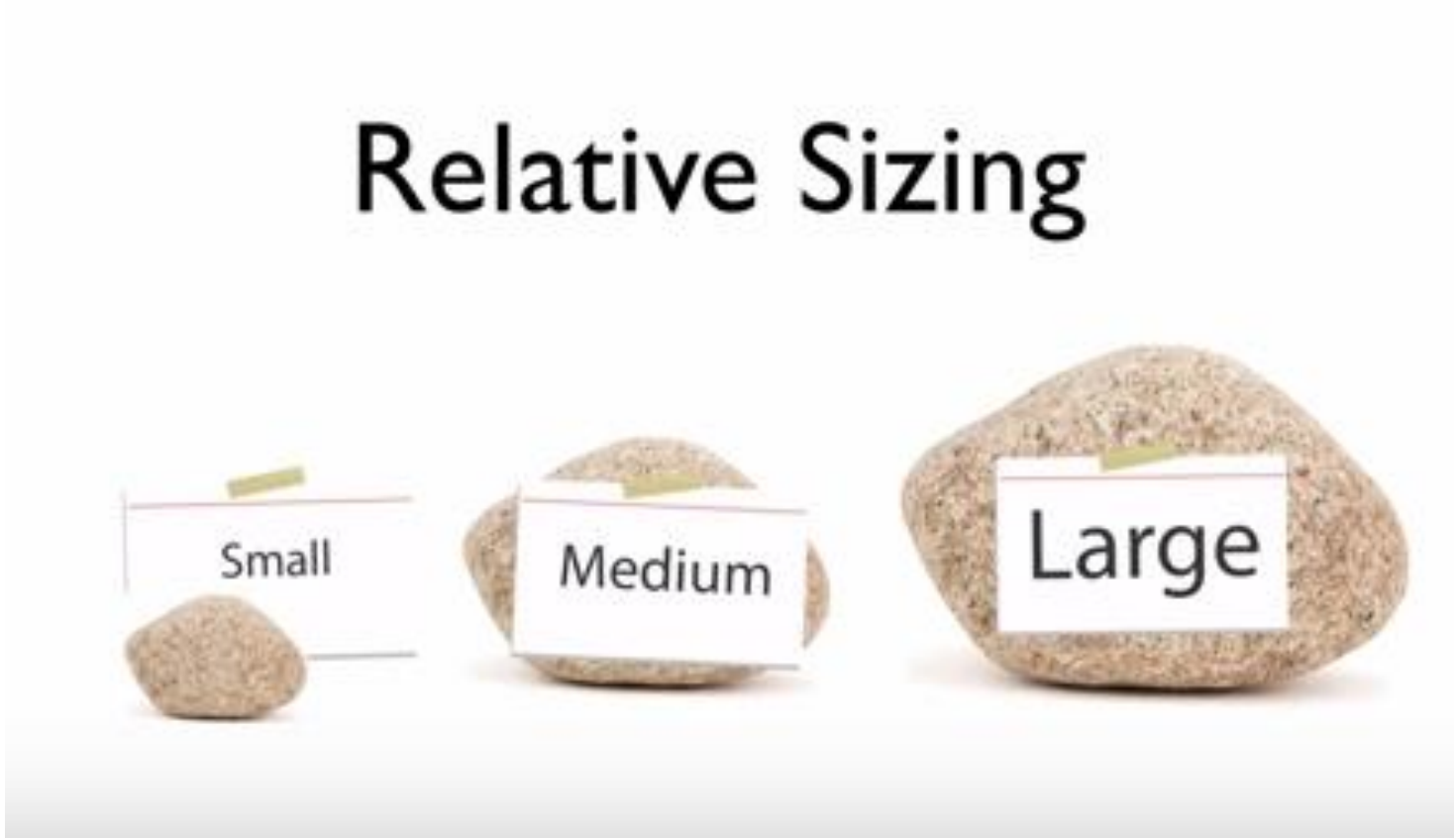


Estimation


Estimate absolutely
X cm²
Y cm²


Estimate relatively
How much bigger

SCRUM



SCRUM

Estimating Stories

- **Story Points**
  - Size matters, not duration
  - Story points are relative
  - Size can be based on several factors

- **Ideal Days**

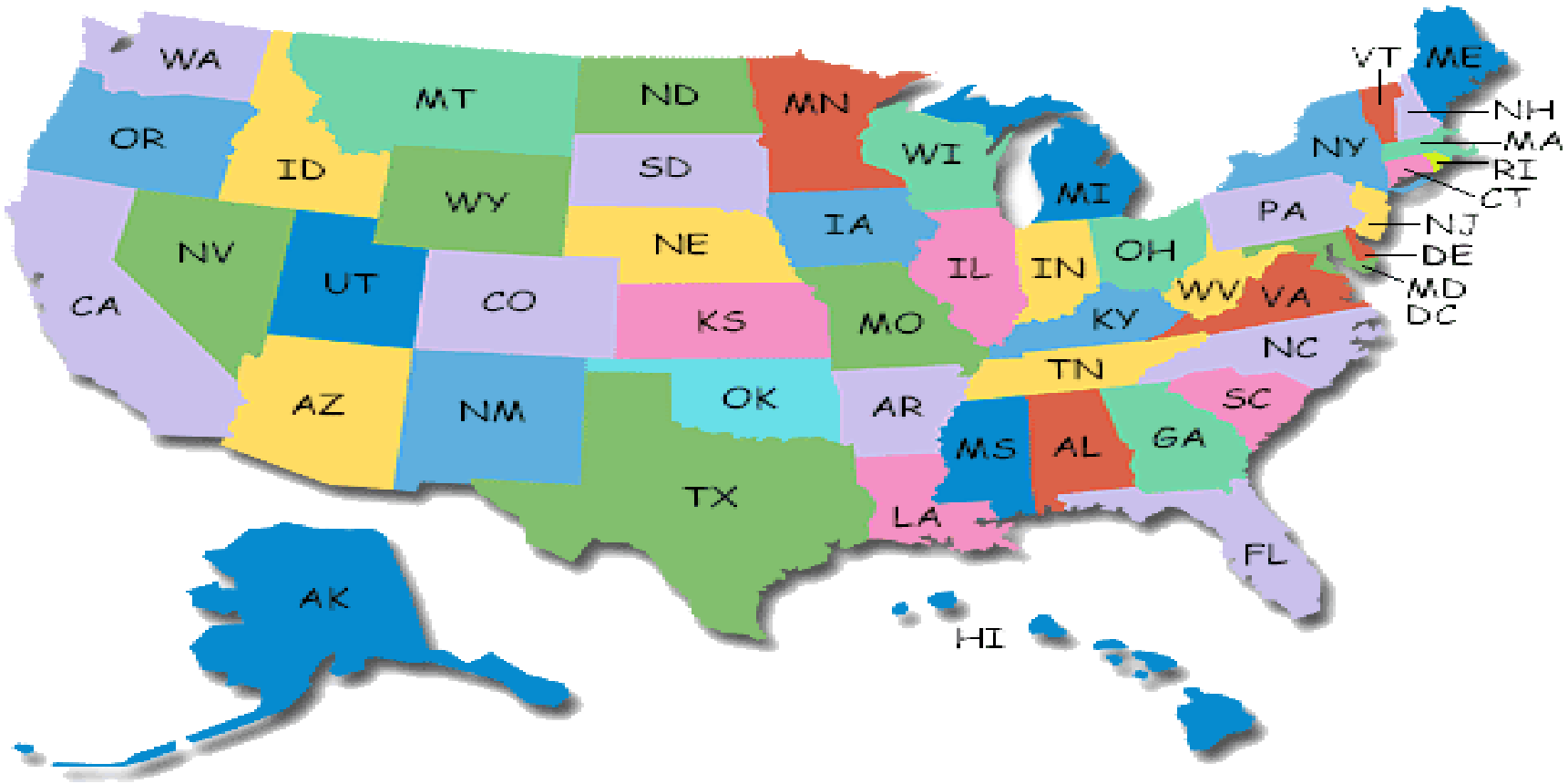
Assigning Story Points

- **T-Shirt Sizing (S, M, L)**
- **Exponential (1, 2, 4, 8)**
- **Fibonacci series (1, 2, 3, 5, 8)**

SCRUM

Relative Sizing

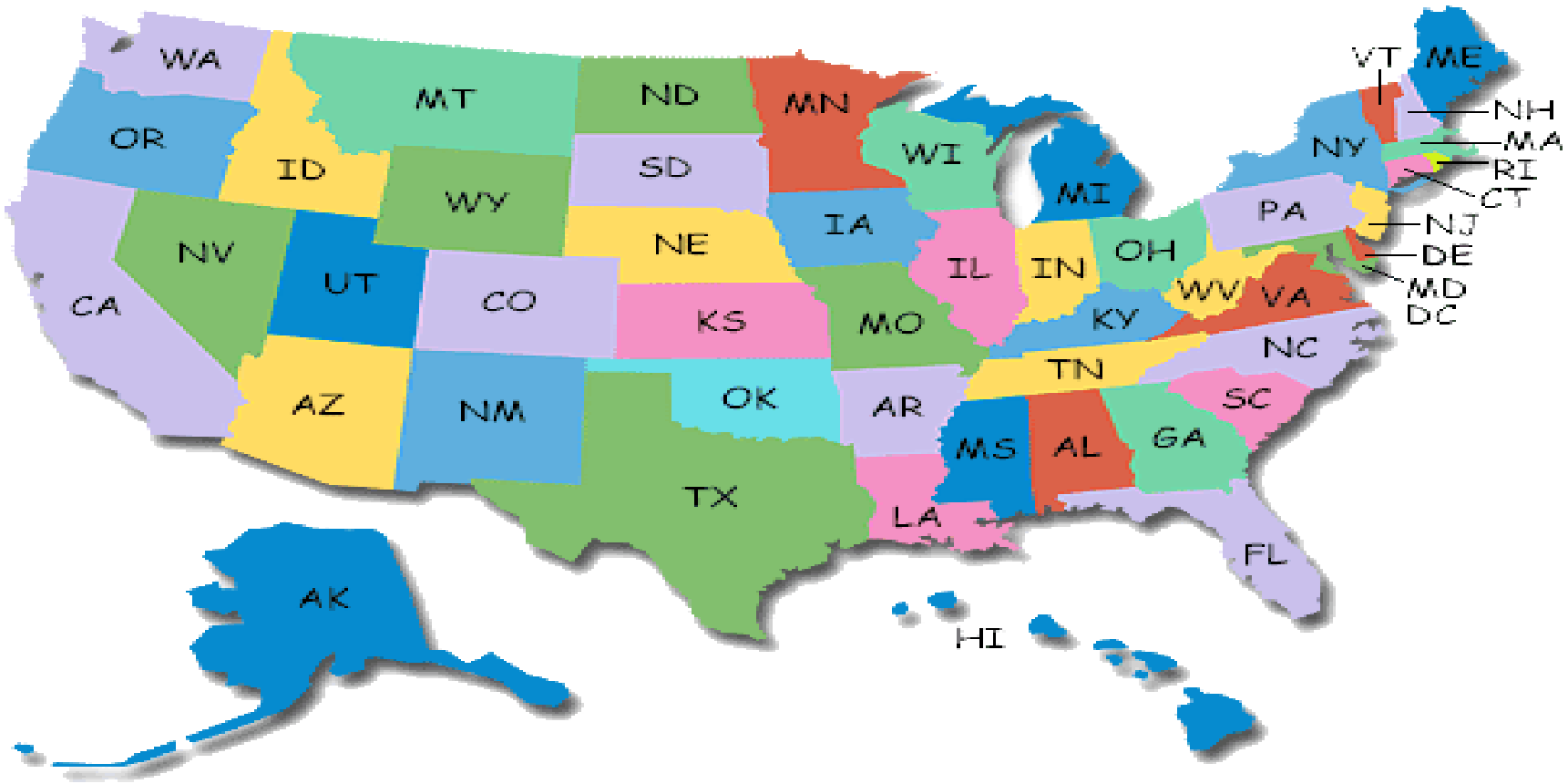
Ohio
Texas
New Mexico
California
New Jersey
Alaska



SCRUM

Relative Sizing

1 – Ohio
8 – Texas
2 – New Mexico
3 – California
1/2 – New Jersey
8 - Alaska



Estimating using Story Points

- **Relative complexity**
- **How long will story x take compared to story y?**
- **Still an estimate**
- **More thorough than other methods**
- **Takes into account productivity / efficiency of the team**

SCRUM

Relative complexity

1 SP

2 SP

3 SP

5 SP

8 SP

13 SP

20 SP

40 SP

100 SP

?

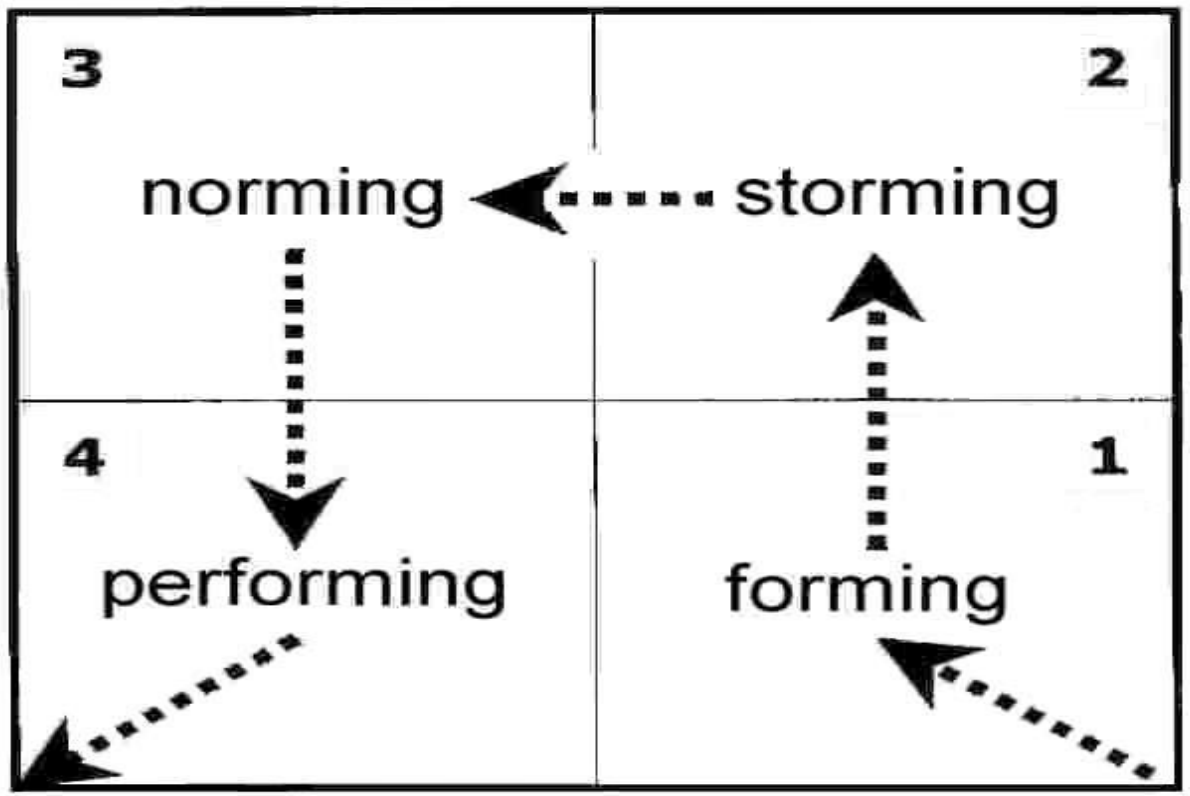
Simple velocity

- **3 simple wooden bridges in 1 sprint**
  - Velocity = 3 story points

- **Alternatively:**
  - 1 simple wooden bridge and 1 basic concrete bridge
  - 1 covered wooden bridge

- **Team velocity increases and decreases**
  - New team members, change in environment etc.

Tuckman: Team development phases



Planning Poker

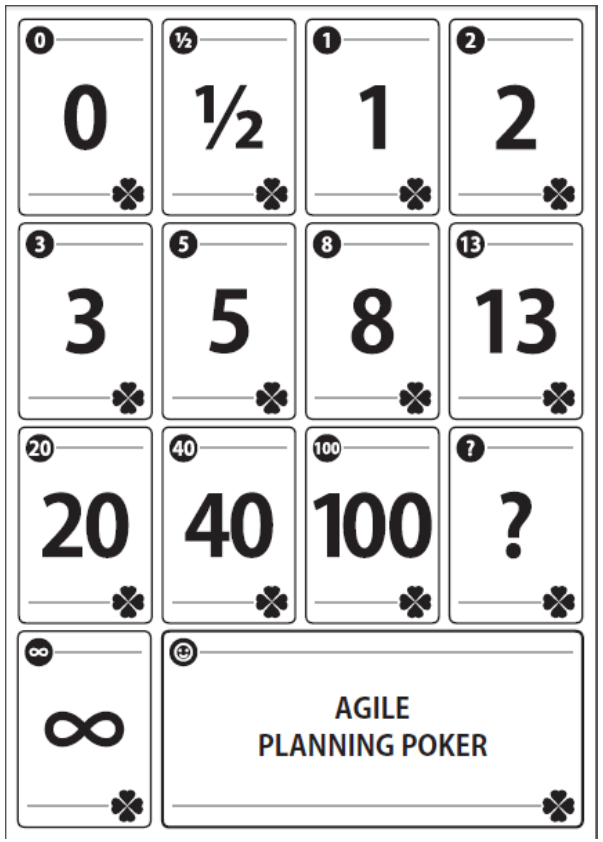
- ***The best way I've found for agile teams to estimate is by playing planning poker (Grenning 2002)***
- **This method tries to make the meetings more short and productive, by making them more fun and dynamic.**

Planning Poker

- **Moderator: Reads Story Description**
- **Ask moderator questions**
- **Pick estimate card**
- **Show cards simultaneously**
- **Discuss Low / High estimates**
- **Consensus (OR Pessimist wins!)**
- **No bluffing**

SCRUM

Preparing the meeting



SCRUM

The Meeting

- A deck is given to each of the members
- The moderator exposes a user story in no more than 2 minutes
- Time for questions about the user story
- Each of the members choose a card privately
- Once everybody has chosen, all the cards are turned over at the same time
- In this first round, it’s probably that the estimations will differ significantly
- In case the estimations differ, the high and low estimators expose their reasons
- A few minutes for the team to discuss about the story and the estimation
- Again, each member thinks privately a estimation, and they show the cards simultaneously
- If the estimations still differ, the same process can be repeated

The Meeting

- **When the estimations converge, the process finishes and the next user story is estimated.**
- **In case the estimations don’t converge by the 3rd round, there are some options:**
  - **Left the user story apart and try again later.**
  - **Ask the user to decompose the story in smaller parts.**
  - **Take the highest, lowest or average estimation.**

SCRUM

Example

- **User story**
  - **As a Salesman , I want to be able to create order to sell products to my customer**

- **Team of 7 members.**
- **First round:**

SCRUM

Example

- **3rd and 6th members expose their reasons for their estimations.**
- **2nd round:**

SCRUM

Example

- **All members have converged except for the 3rd**
- **A new round of expositions and voting can be made.**
- **It's also possible to take 3 or 5 as the estimation.**

Conclusion

- **Multiple expert opinions.**
- **The dialogue between the members result in more accurate estimations.**
- **Studies have shown that averaging estimations and group discussion lead to better results.**
- **It’s fun!**
- **Meetings with all the team are expensive.**
- **The moderator needs to be careful and control the meeting so it doesn’t get too long.**
- **Some factors can interfere in the estimations: dominant personalities, company politics…**
- **Discussions can end in polarized estimations.**

Exercice

- **Estimate with planning poker**
- **Product backlog**

Exercise 1 – Countries

- **Spain**
- **China**
- **Luxemburg**
- **Denmark**
- **South Africa**
- **India**
- **Colombia**
- **Portugal**

- **Advice 1 : Take one country as a ruler value 1**
- **Advice 2 : Take Portugal as 1**

Exercise 1 – Countries

Country	Card 1	Card 2	Card 3
Spain			
China			
Luxemburg			
Denmark			
South Africa			
India			
Colombia			
Portugal			

Exercise 1 – Countries

Country	Card 1	Card 2	Card 3	Correct
Spain				5
China				100
Luxemburg				0
Denmark				1/2
South Africa				13
India				40
Colombia				8
Portugal				1

Exercise 2 – Animals

- **Impala**
- **Elephant**
- **Giraffe**
- **Rat**
- **Crocodile**
- **Snake**
- **Dog**

Exercise 2 – Animals

Animal	Card 1	Card 2	Card 3
Impala			
Elephant			
Giraffe			
Rat			
Crocodile			
Snake			
Dog			

Exercise 2 – Animals

Animal	Card 1	Card 2	Card 3	Correct
Impala				3
Elephant				100
Giraffe				40
Rat				0
Crocodile				8
Snake				½
Dog				1

Exercise 3 – vehicule size

- **Convertible**
- **Motorbike**
- **Starship Enterprise**
- **SUV**
- **Minivan**
- **Bus**

Exercise 3 – vehicule size

Vehicule	Card 1	Card 2	Card 3	Correct
Convertible				
Motorbike				
Starship Enterprise				
SUV				
Minivan				
Bus				

Exercise 3 – vehicule size

Vehicule	Card 1	Card 2	Card 3	Correct
Convertible				3
Motorbike				1
Starship Enterprise				100
SUV				5
Minivan				8
Bus				20

Exercise 4 – Product backlog – Web site development

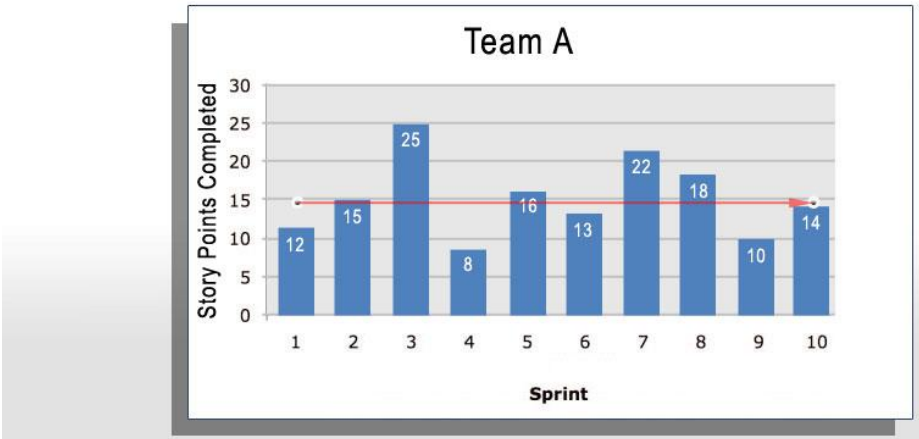
- 1.Logon entry screen
- 2.Validation of userid and password
- 3.Password forgotten module
- 4.HTML Web page with formatted text
- 5.HTML Web page with animations
- 6.HTML Web page with photos
- 7.HTML Web page with video
- 8.Entry screen and creation of a database record
- 9.Validations with reference table on a one field
- 10.Search a list of records
- 11.Retrieve a record and to update data information
- 12.Delete a record with validation and confirmation screen
- 13.Report a list of records
- 14.Graphs of data metrics

Exercise 4 – Change Request

- •15.Payment module by credit card
- •16.Keywords data entry to identify communities

What is velocity ?

- Velocity is the measure of the productivity of the team
- The rate at which the team is completing work from the product backlog
- Velocity is the most important metric as it tells us how much work can be done in a sprint
- Velocity can be used in ideal time or story point format.



How to calculate initial velocity ?

- **When we start a project, we do not know what our velocity is, we have to somehow work out an initial estimated velocity that we think we can achieve in the first sprint. When we are bit into our first sprint we will begin to realize our velocity as features begin to be completed.**
- **The following options can be used to calculate an initial velocity:**
  - **If you are lucky to have previous projects that have used scrum then use their actual velocity for similar work and team skill or composition. If it is the same team then even better**
  - **Predict the velocity, take some stories, expand them into tasks to work out what you think you can complete**
  - **Execute a short sprint to complete a few stories to work out velocity**

SCRUM

Sprint 1

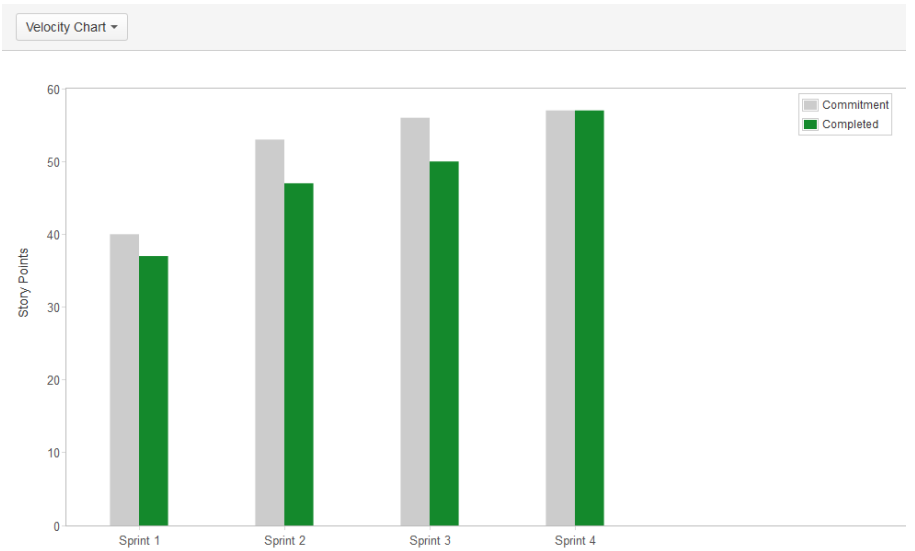
- **Developers will commit to XX story points**
- **Warning, they will usually overcommit**
- **After the end of sprint one, you have your first velocity number**

Estimate an Initial Cost and Duration

- **Cost of Project = Money Spent + (Sprints Remaining X Sprint Burn Rate)**
- **Sprints of 20 business days**
- **Estimate of our Product Backlog 90 Story points**
- **We determine that our initial velocity for sprint 1 is 30**
- **How many sprints do we need ?**
- **Duration = Size Remaining / Velocity Duration = 90 / 30 Duration = 3 Sprints**
- **At a high level we know that our team will roughly burn around $20,000 per month**
- **Cost of Project = $0 + (3 X $20,000) = $60,000**

Re-estimation

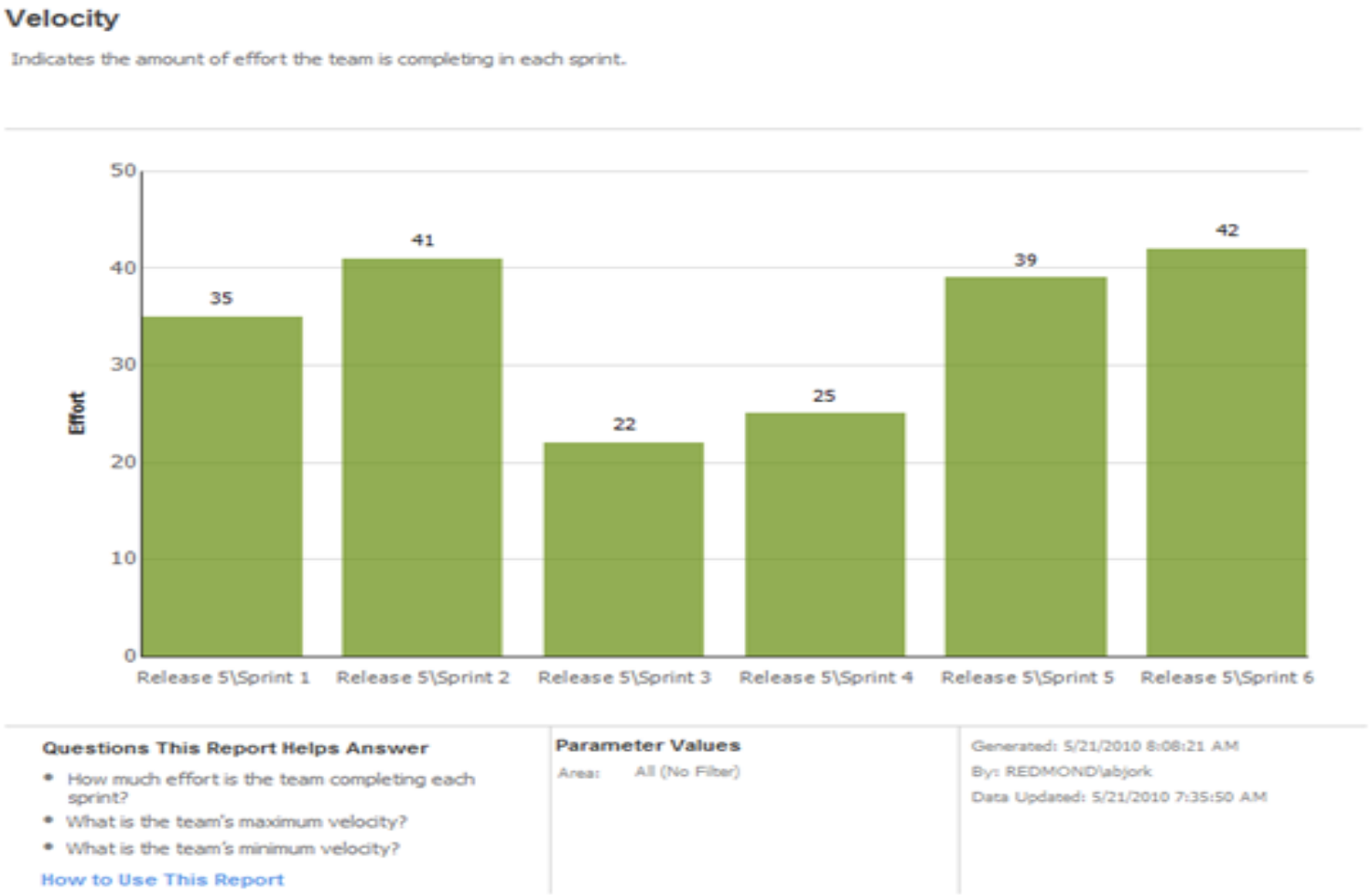
- **As you complete more sprints, your velocity will change**
  - **Velocity changes because of minor inconsistencies in the story point estimates**
  - **Team velocity will typically stabilize between three and six iterations**
- **Re-estimation of the entire project happens after each sprint**
  - **New velocity**
  - **New story points added and removed (completed)**



Sprint	Commitment	Completed
Sprint 1	40	37
Sprint 2	53	47
Sprint 3	56	50
Sprint 4	57	57

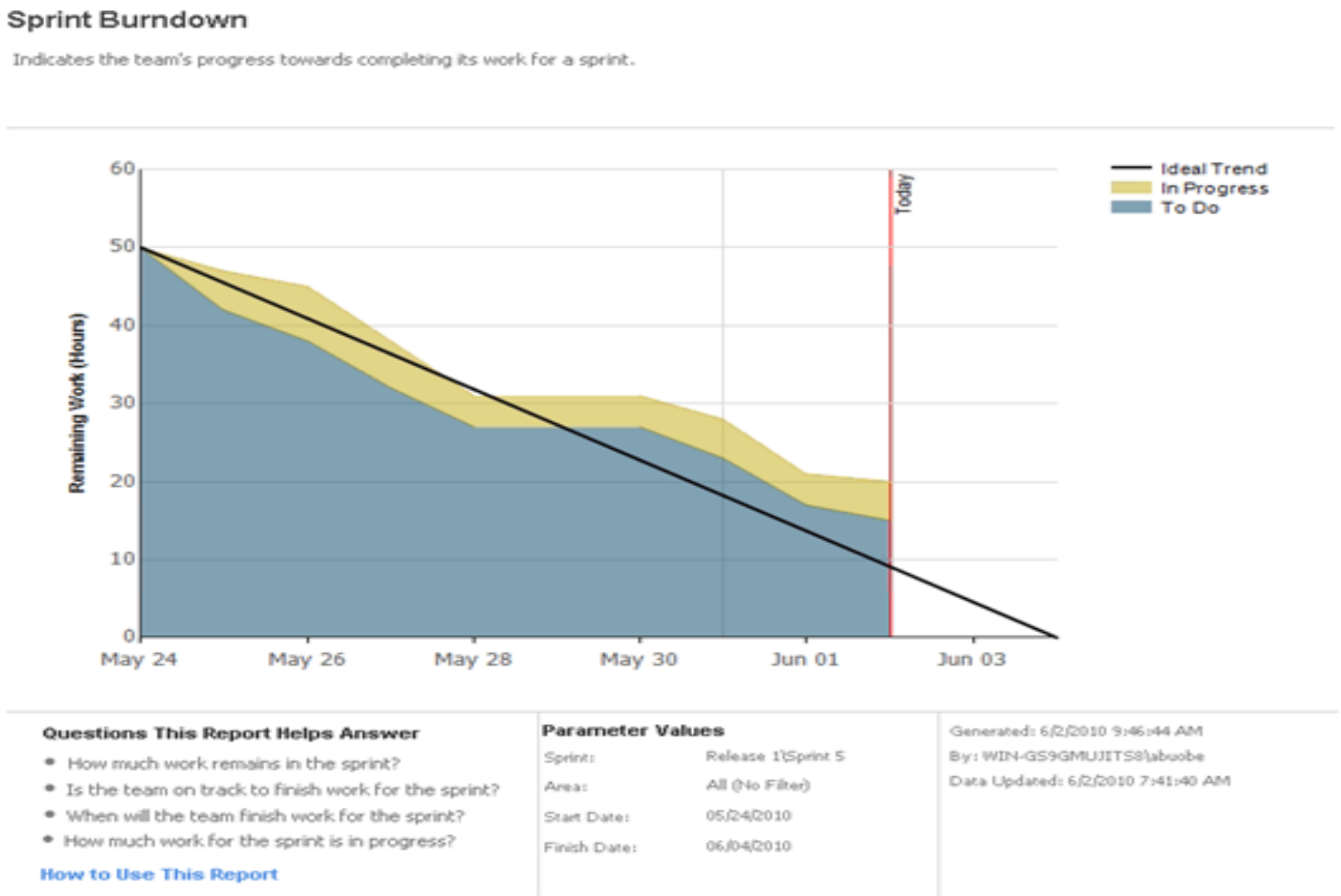
SCRUM

Velocity



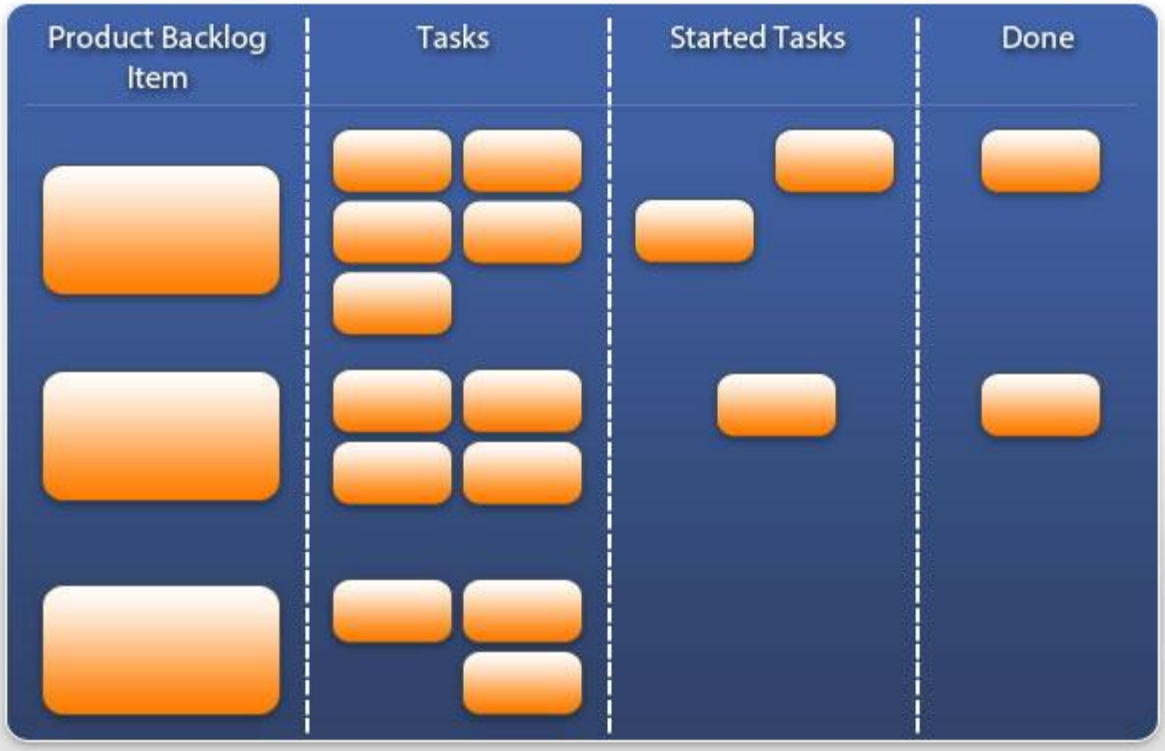
SCRUM

Sprint Burndown Chart



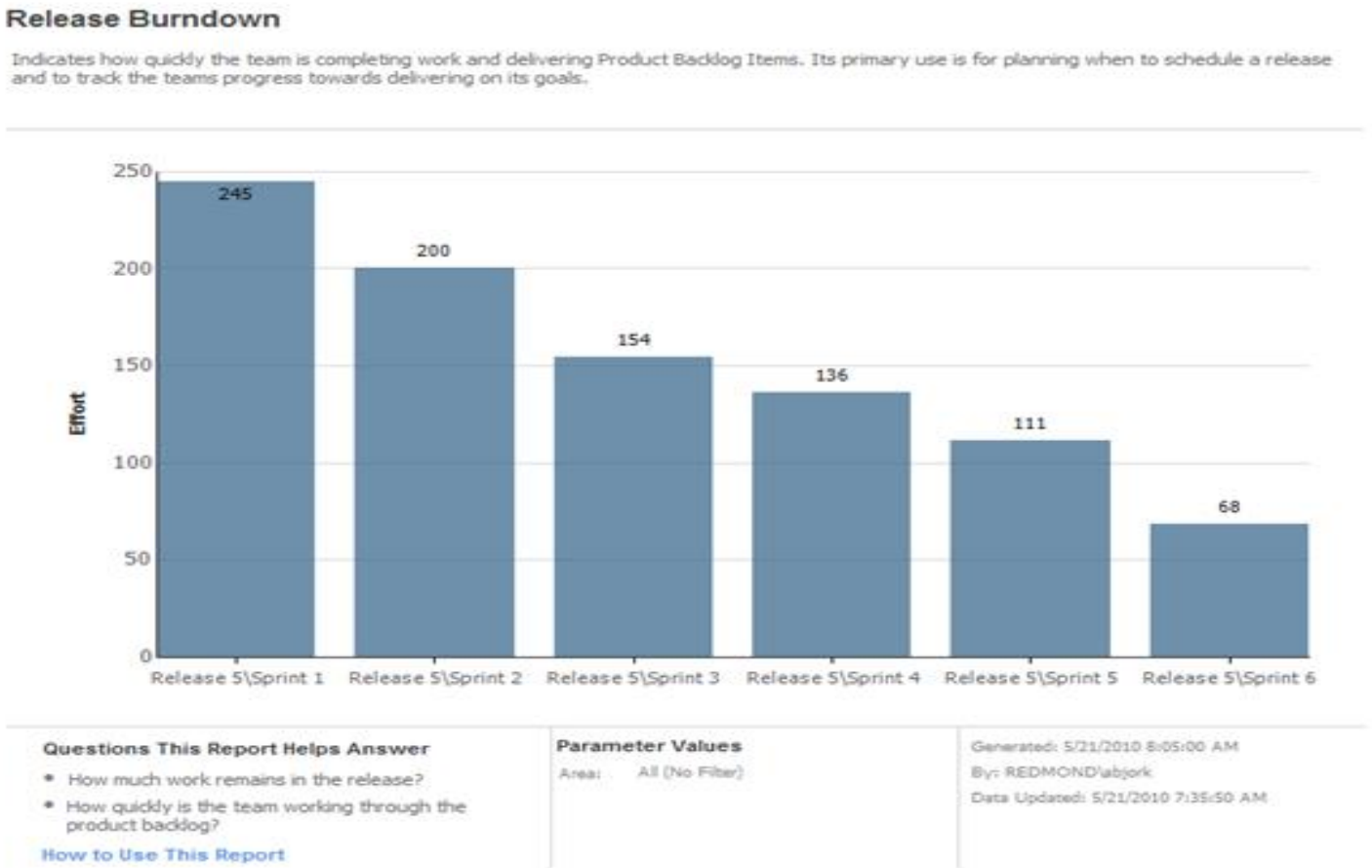
- **Can be used in the daily stand up**

Agile planning



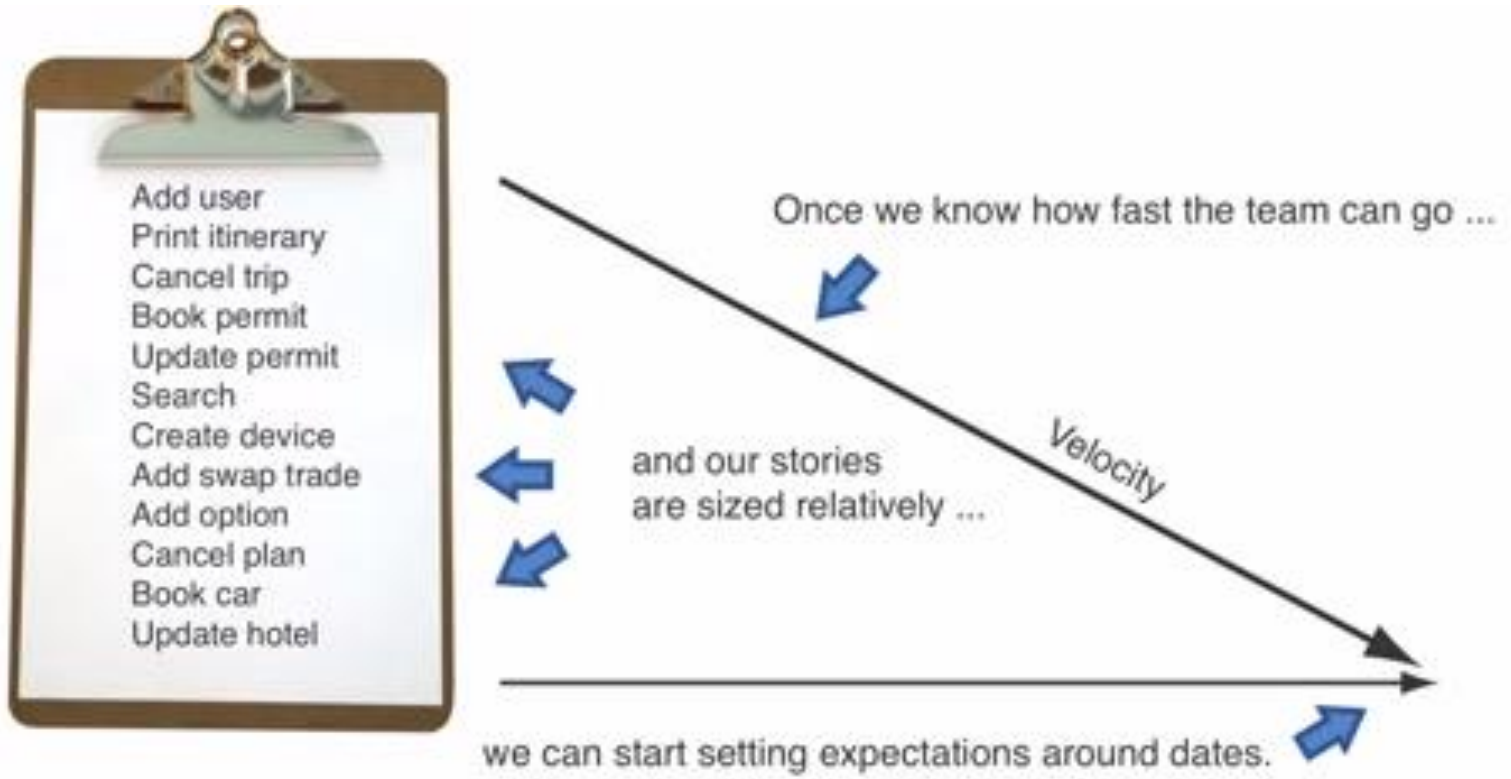
SCRUM

Release Burndown Chart

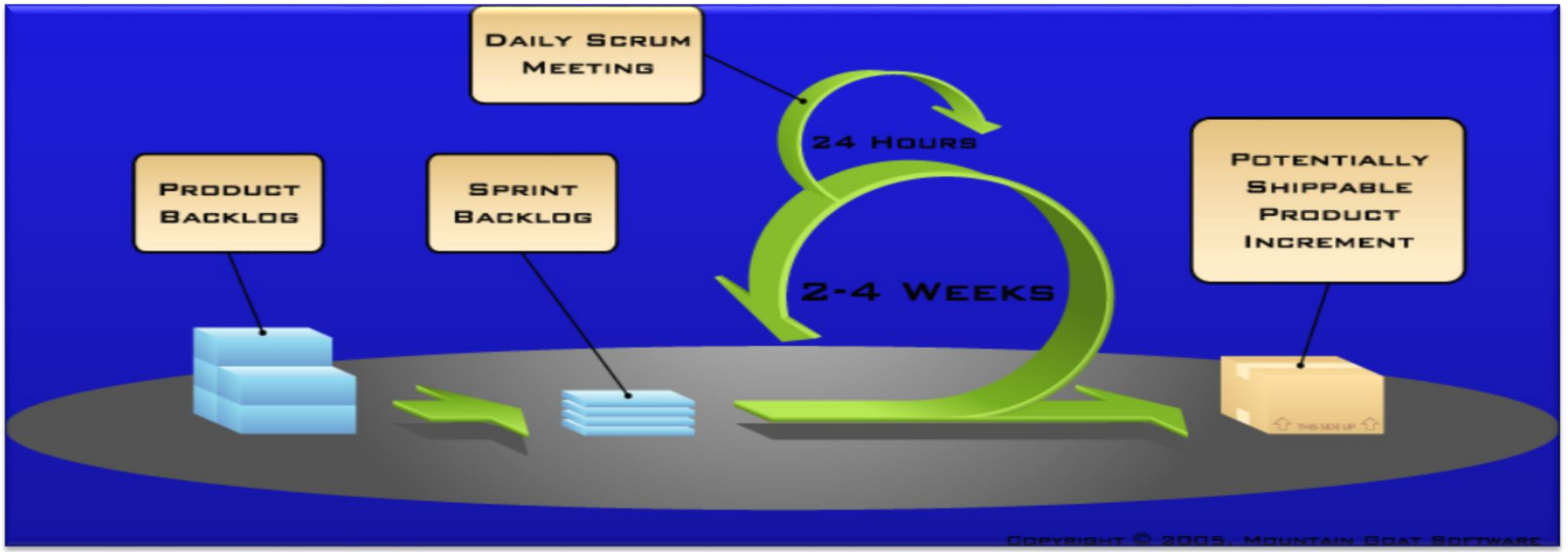


SCRUM

Agile planning



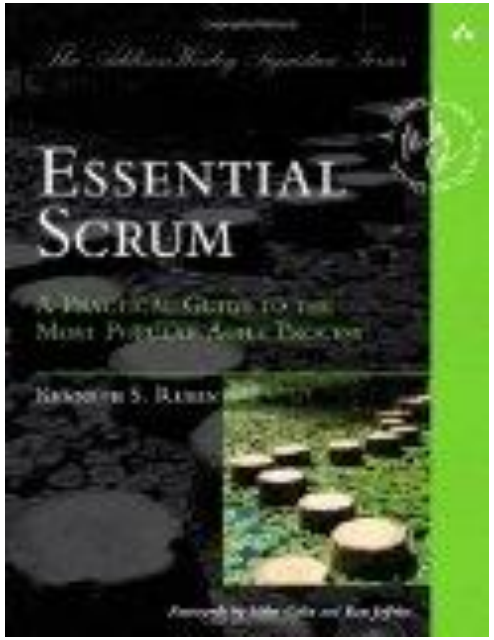
SCRUM

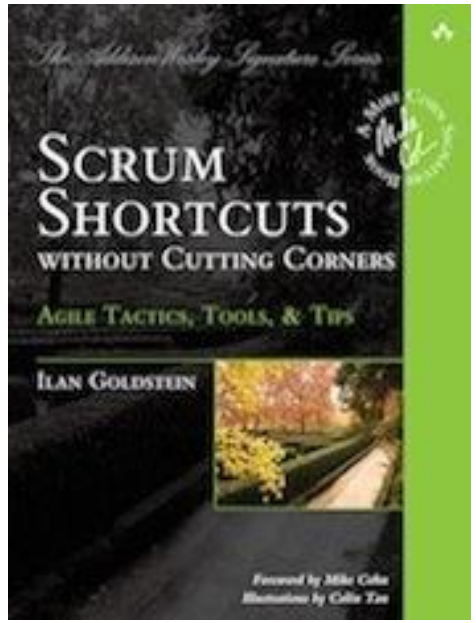


EPITA
ÉCOLE D'INGÉNIEURS EN INFORMATIQUE

SCRUM

Reference books





Sources and Bibliography

- Agile Scrum Introduction, Đức Quốc
- Essential Scrum, A practical guide to the most popular agile process, Kenneth s. Rubin, Addison Wesley
- Agile and Iterative Development: A Manager's Guide by Craig Larman
- Agile Estimating and Planning by Mike Cohn
- Agile Project Management with Scrum by Ken Schwaber
- Agile Retrospectives by Esther Derby and Diana Larsen
- Agile Software Development Ecosystems by Jim Highsmith
- Agile Software Development with Scrum by Ken Schwaber and Mike Beedle
- Scrum and The Enterprise by Ken Schwaber
- Succeeding with Agile by Mike Cohn
- User Stories Applied for Agile Software Development by Mike Cohn

SCRUM

Webography

- http://www.scrum-institute.org/Example_Scrum_Certification_Test_Questions.php
- http://coach-agile.com/wp-content/uploads/2013/04/Lego4Scrum-version2-French.pdf
- http://www.agiliste.fr/guide-de-demarrage-scrum/
- www.mountaingoatsoftware.com/scrum
- www.scrumalliance.org
- www.controlchaos.com
- http://creativecommons.org/licenses/by/3.0/