Agile & more...

ELAD SOFER - AGILE COACH

HTTP://WWW.PRACTICAL-AGILE.COM/OUR-BLOG
ELAD@PRACTICAL-AGILE.COM

TWITTER: @ELADSOF

Practical Agile
Agile in your Context

PLEASE SILENCE YOUR CELL PHONES BEFORE SERVICE BEGINS

THANK YOU

SHHH...


Parking lot


WELCOME TO
HARD WORK
STAY STRONG

Course structure

Sunday	Agile & Scrum Plan and prepare product
Monday	NodeJS Build some server
Tuesday	AngularJS Build some frontend
Wednesday	ElasticSearch Add the DB
Thursday	Finalize products Summary

Practical Agile
Agile in your Context

Exercise

Who am I ?

Tell us something interesting…

Practical Agile
Agile in your Context

About me

- Software developer
- Agile coach
- LeSS certified trainer
- Father and husband
- Amateur DJ

Practical Agile
Agile in your Context

Who am I ?

- S/W Engineer.
- 18 years in s/w (embedded & applications)
- Agile coach.
- Blogger
- Co-Founder of “Practical-Agile”
- Co-Organizer of “Agile Practitioners IL” group

Practical Agile
Agile in your Context


Requirements
Analysis
Design
Implement
Test
Acceptance
Deliver
Practical Agile
Agile in your context

The waterfall development model originates in the manufacturing and construction industries

The first description of waterfall is a 1970 article by Winston W. Royce

Royce presented this model as an example of a flawed, non-working model

"I believe in this concept, but the implementation described above is risky and invites failure" [Royce 1970]

The harder we plan and analyze in the beginning, the less there's change in the project and the more successful the project.	There is change always and responding to it is vital. Uncertainty is best reduced by learning from actual implementation
It is possible to "collect" or even "know" all the requirements up-front	Requirements evolve as customers and our knowledge increases – based on experience
Division of work to specialized teams (specification, design and testing) is efficient	Cross-functional teams reduce the amount of handovers and delays, thus they are more productive
Product development process can be defined as a predictable and repeatable process	Product development is an evolving and adaptive process, unique for every organization.

Agile in your Context

Wishful thinking

No matter how hard you try, it ain't gonna work

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

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.

Agile Principles 1/2

1. Our highest priority is to **satisfy the customer** through **early and continuous** delivery of valuable software
2. Welcome changing requirements, even late in development. **Agile processes harness change** for the customer's competitive advantage.
3. Deliver **working software frequently**, from a couple of weeks to a couple of months, with a preference to a shorter timescale.
4. Business people and developers must work **together daily** throughout the project.
5. Build project around motivated individuals. Give them the **environment and support** they need, and **trust them** to get the job done.
6. The most efficient and effective method of conveying information to and within development team **is face-to-face conversation**.

Practical Agile
Agile in your Context

Agile Principles 2/2

7. Working software is the **primary measure for progress**.
8. Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to **technical excellence** and good design enhances agility.
10. **Simplicity – the art of maximizing the amount of work not done – is essential.**
11. The best architectures, requirements, and designs emerge from **self-organizing teams**.
12. At **regular intervals, the team reflects** on how to become more effective, then tunes and adjusts its behavior accordingly.

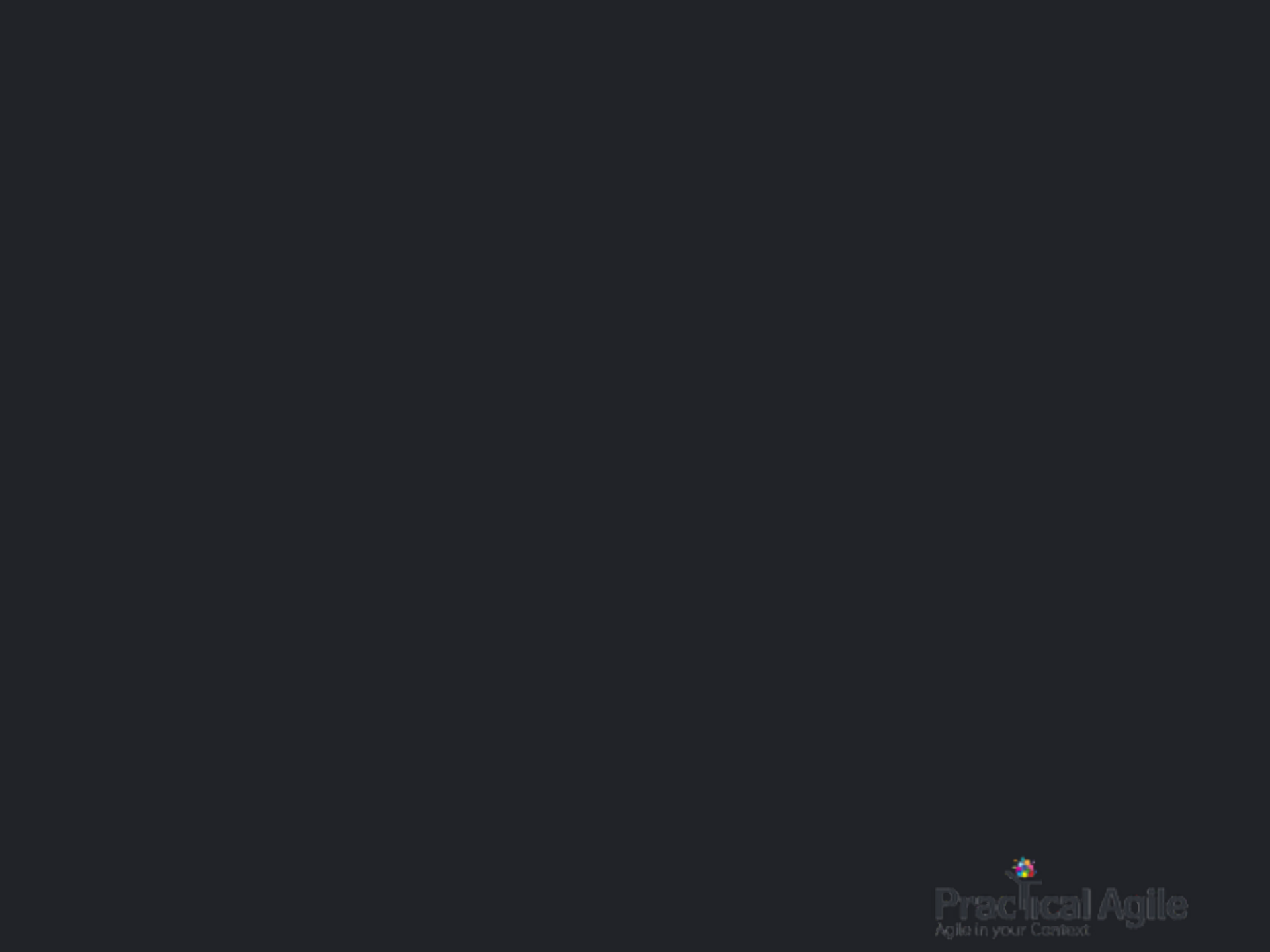
Practical Agile
Agile in your Context

Practical Agile
Agile in your Context

Principles behind Scrum

- Understanding that we cannot predict the future.
- One size **does not** fit all.
- Constant improvement.
- Transparency & Visibility
- Team work
- Deliver business value fast (max. 30 days)
- Prioritizing – Industry statistics show: 65% of all features are rarely\never used.
- Empirical approach

Practical Agile
Agile in your Context


Practical Agile
Agile in your Context

Scrum process overview

Practical Agile
Agile in your Context

spiral (waterfall) vs. iterative

Practical Agile
Agile in your Context

Roles

- Product owner
- ScrumMaster
- Team

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Roles

- Product owner
- ScrumMaster
- Team

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Scrum product owner

- Defines the features of the product
- Defines release dates and content
- Responsible for ROI.
- Prioritizes features.
- Can change features and priority once every predefined interval.
- Decides what will be worked on in each iteration
- Accepts or rejects results.
- Work with the team - **DAILY!**

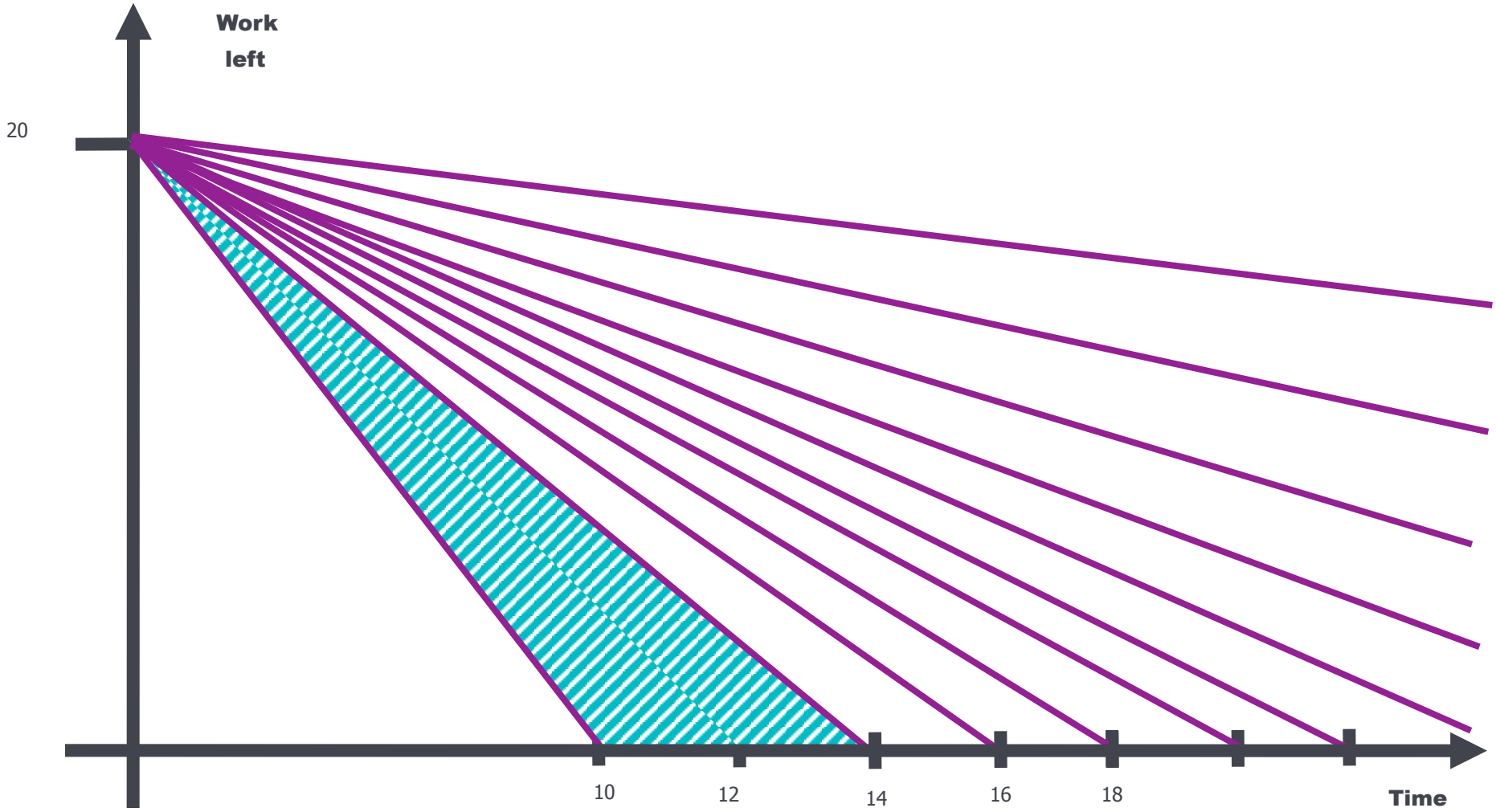
Practical Agile
Agile in your Context

The scrum master.

- Responsible for the scrum process.
- Protects the team.
- Helps removing impediments.
- He is standing at the nexus between:
  - The product management that believes that any amount of work can be done.
  - Developer's that have the willingness to cut quality to support the managements belief.

Practical Agile
Agile in your Context

Management vs. Development – Tech debt



Practical Agile
Agile in your Context

Exercise

Is command and control Management?

The team

- Typically 5-9 people
- Cross-functional:
  - "Architects", Programmers, testers UI designers, etc.
- Members should be full-time
  - May be exceptions (e.g., database administrator)
- Teams are self-organizing
  - Ideally, no titles but rarely a possibility
- Membership should change as little as possible
  - only between sprints

Practical Agile
Agile in your Context

Roles

- Product owner
- ScrumMaster
- Team

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Product backlog

- List of features, Technology, issues.
- Items should deliver value for customer.
- Constantly prioritized & Estimated.
- Anyone can contribute.
- Visible to all.
- Derived from business plan, may be created together, with the customer.
- Can be changed every sprint!!!
- Customer is not “programmed” to think of everything in advance.

Practical Agile
Agile in your Context

Product backlog example

Backlog item	Estimate
As a user I would like to register	3
As a user I would like to login	5
As a buyer I would like to make a bid	3
As a buyer I would like to pay with a credit card	8
As a seller I would like to start an auction	8
...	...
Test register feature	13
Create infrastructure for login	20

Practical Agile
Agile in your Context

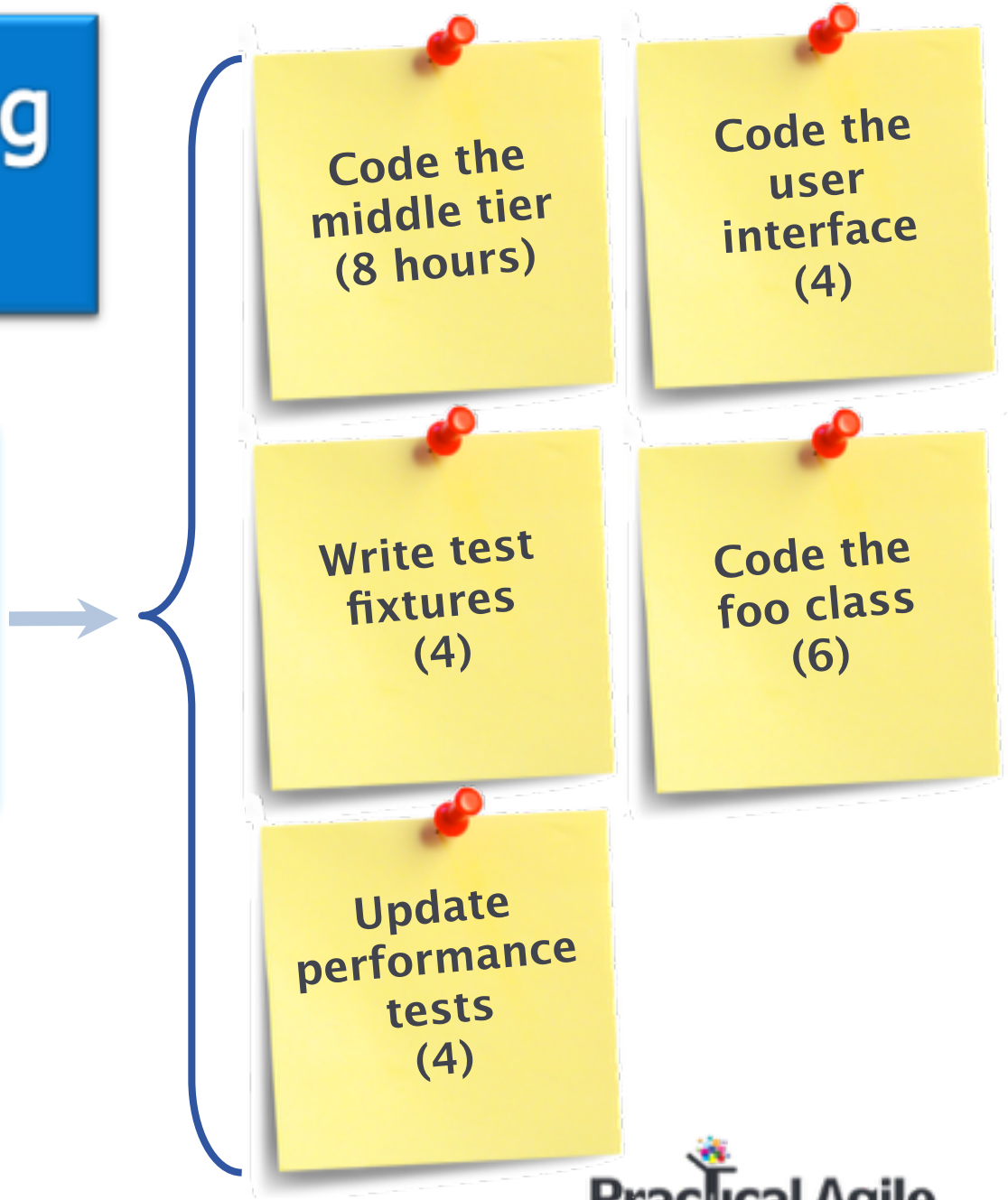
Sprint Backlog

- The sprint backlog is defined by understanding and agreeing on the sprint goal(s) and selecting the appropriate items from the product backlog.
- The goal is determined by the customers\product owner \team.
- The team compiles a list of tasks that are needed in order to complete the sprint goal(s).
- A task should be as small as possible and should not exceed a time period of 2 days (time not effort).
- If a task X can not be defined, there will be a task to define the task X.
- The sprint backlog can be modified throughout the sprint.

Practical Agile
Agile in your Context

Sprint Backlog - Example

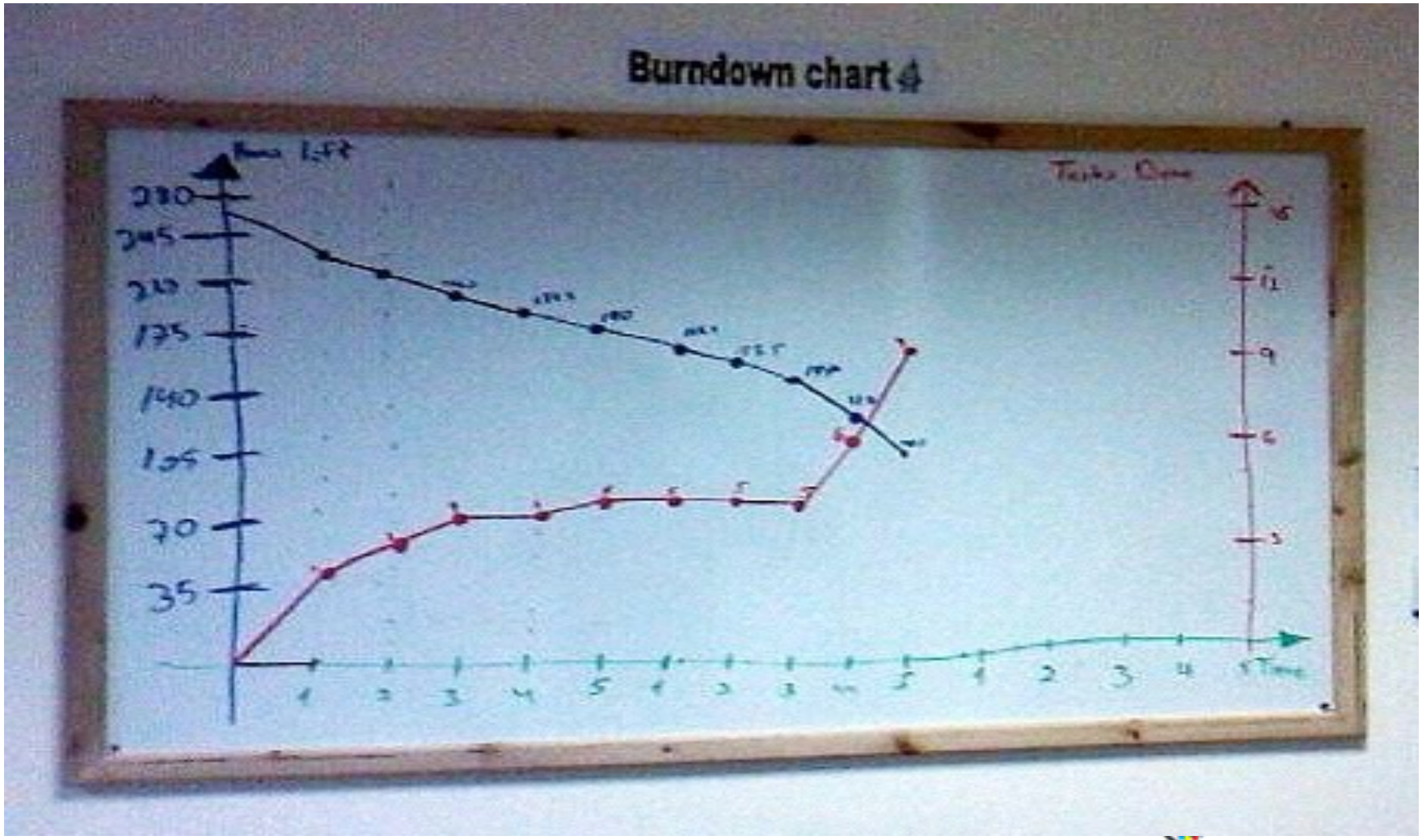
As a user I would like to register



Practical Agile
Agile in your Context

Sprint Backlog – Real-life example

Burndown chart – real-life example



Practical Agile
Agile in your Context

Roles

- Product owner
- ScrumMaster
- Team

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Sprint planning

- Usually the longest meeting of all.
- The meeting takes place prior to every sprint.
- Participants:
  - All Team members , PO, Scrum master.
- Is divided into two Parts:
  - Part I
    - Team and PO discuss and clarify the top priority items
    - Team and PO selects sprint Goal.
  - Part II
    - Team creates the sprint backlog
    - Team commits on content of coming sprint.

Practical Agile
Agile in your Context

Sprint review meeting

- At the end of each sprint there is a meeting called the sprint review.
- The purpose of the meeting is to let the "captain" to know where the ship is heading and where it is in it's route. In addition all new features will be presented to the product owner.
- During this meeting the team presents to the management \customers\users\product owner, what work has been **DONE** and what was not.
- The only form of "automated" presentations allowed is working software, Slideware is banned.
- The things that were not accomplished will be returned to the product backlog.

Practical Agile
Agile in your Context

What is "DONE" ?

- We have in Scrum – DOD – Definition of Done.
- Terms of satisfaction of the PO
- Only DONE items count
- Success is well defined
- Example:
  - Unit tested, Verification, Documented, deployed.

Practical Agile
Agile in your Context

Daily scrum.

- A meeting that occurs daily at the same time. **anyone** who wants attend , can do so.
- Each of the team members needs to answer briefly these three questions:
  1. What have you done since the last daily scrum?
  2. What will you do until the next daily scrum?
  3. What got in your way of doing work?
- The team does not report to anyone but the team.
- During the meeting only one of the team members is allowed to speak, others should keep quiet.
- All of problems raised in the meeting should be written down and resolved by the scrum master\team.
- The daily scrum is **not** a technical meeting.

Practical Agile
Agile in your Context

Sprint retrospective

- Periodically take a look at what is and is not working
- Typically takes ~60 minutes
- Done after every sprint
- Whole team participates
  - Scrum Master
  - Product owner
  - Team
  - Possibly customers and others

Practical Agile
Agile in your Context

Start / Stop / Continue

Whole team creates "experiments" they'd like to try

Start doing

Stop doing

Continue doing

Practical Agile
Agile in your Context

The sprint

- The sprint is the productive part of the scrum
- It is a fixed, predefined, period of time.
- During this time the work load, the scope or nature of work must not be changed. The only "manager" of the scope is the sprint backlog.
- The team is free to accomplish the sprint goal as it see's fit, within the limits of the team's procedures and the time limits.
- During the sprint, the team has total freedom over how it works:
    - Work as many hours as it wants.
    - Hold meetings whenever it wants
  - During the sprint the team is accountable for only two things
    - Daily scrum
    - Sprint backlog.

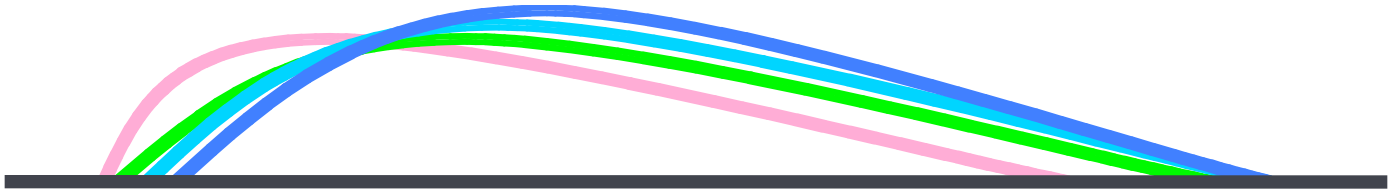
Practical Agile
Agile in your Context

Sequential vs. overlapping development

Design | Code | Integrate | Test

Rather than doing all of one thing at a time...

...Scrum teams do a little of everything all the time



Source: "The New New Product Development Game" by Takeuchi and Nonaka. *Harvard Business Review,* January 1986.

Practical Agile
Agile in your Context

Practical Agile
Agile in your Context

Scrum - Summary

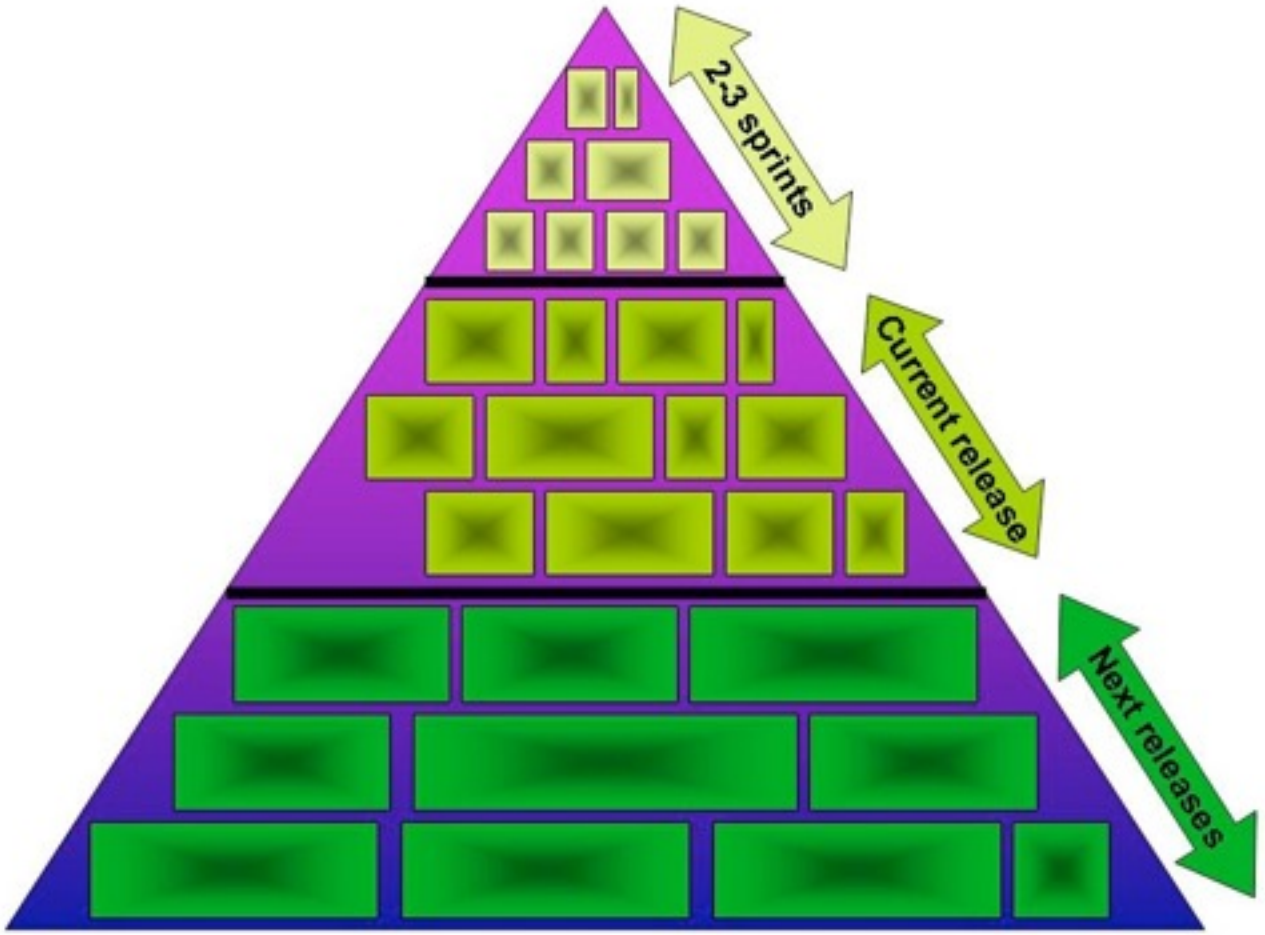
- The framework.
- Roles
- Ceremonies
- Product backlog
- Sprint backlog
- Burndown charts
- Definition of Done.

Practical Agile
Agile in your Context

AGILE ESTIMATION AND PLANNING

Practical Agile
Agile in your Context

The product backlog



Backlog items size

- The product backlog items (PBI) become more granular the higher they are in the backlog.
- Usually the PBI starts as very big items (Epics) and are split into smaller sized items.
- It is common that the PBIs for the next 2-3 sprints are split so that:
  - 3-5 PBIs fit within a single sprint.

Practical Agile
Agile in your Context

Splitting backlog items

- Different scenarios
- Stubbing\mocking external dependencies
- Splitting across the data model.
  - Support only a subset of attributes
- Splitting across operations
  - CRUD \ parts of a protocol
- Splitting on results
  - Success and failure scenarios.
- Splitting cross-cutting concerns
  - Logging \ Security.
- Splitting functional & non-functional requirements

Acceptance Criterion

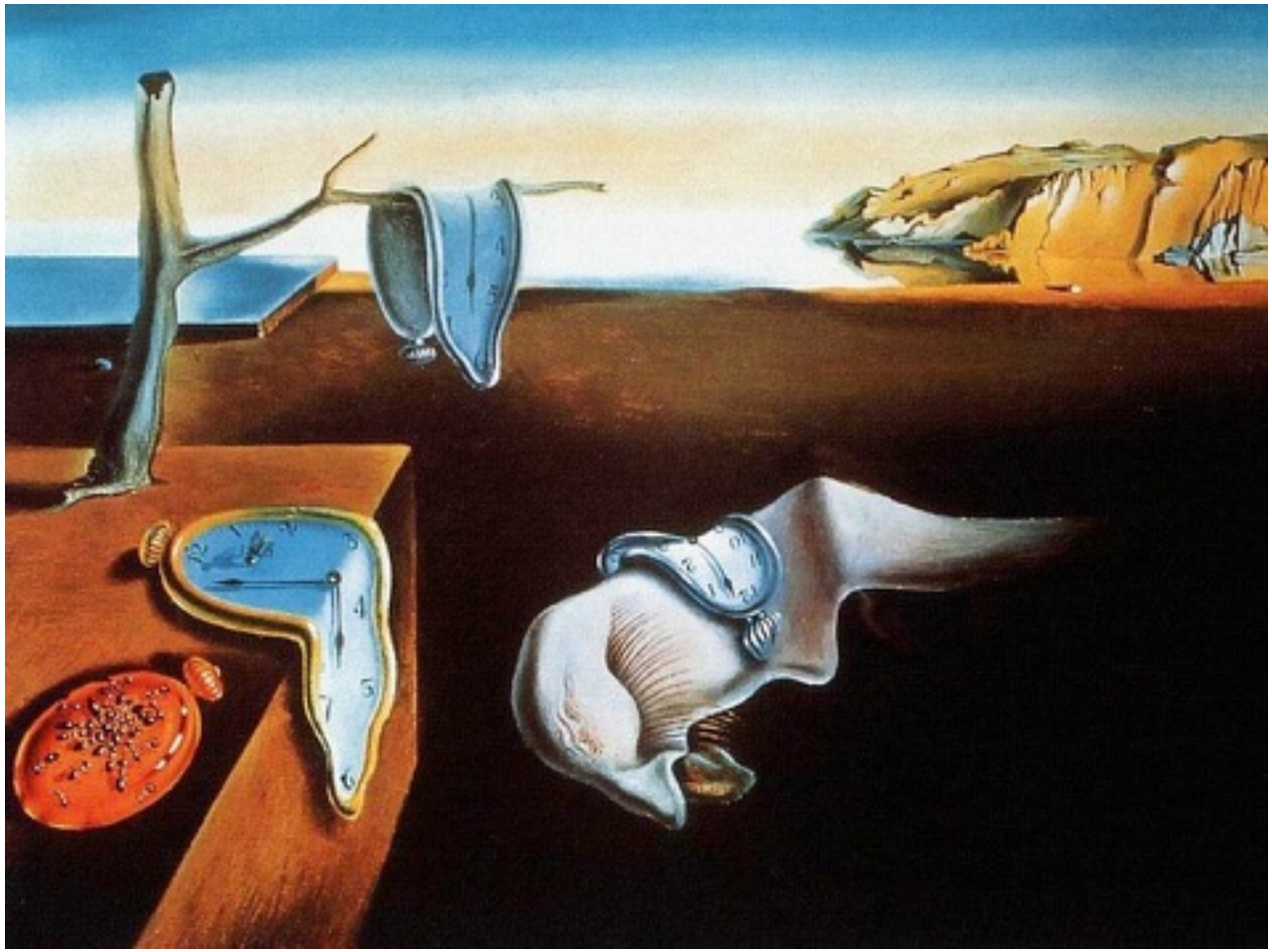
- Recommended format
  GIVEN a pre-condition
  WHEN an action happens
  THEN an expected result occurs
- Some people call it a test
  [And some others resent the notion of tests in requirements]
- Can promote a test-first culture

Practical Agile
Agile in your Context

Estimating

"It's better to be roughly right than precisely wrong."
[John Maynard Keynes[

Persistence of time



Practical Agile
Agile in your Context

Relative estimation

Practical Agile
Agile in your Context

Why relative ?

- We are not good in measuring absolute values.
- We are good in comparing things.
- We have the basic math skills (or a calculator).
- High accuracy has a high toll.
- Estimates become commitments
- Time is not persistent.

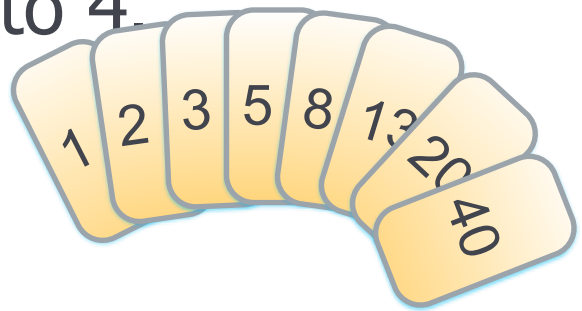
Story points

- Name is derived from user stories.
- They reflect the “bigness” of a user story.
  - How hard it is ?
  - How risky it is ?
  - How much of it there is ?
- Relative values matters.
- Unitless.
- Point values “include uncertainty”.
- Easy and quick
  - A little effort helps a lot
  - A lot of effort helps a little more

Practical Agile
Agile in your Context

Planning poker

- Each person gets a deck of cards (Fibonacci).
- The item to be estimated is read to all.
- Attendants ask clarifications for the item.
- Each person selects a card and puts it on the table facing down.
- When everyone is done, cards are exposed.
- If the estimations do not match a short discussion is done – Highest & Lowest estimators speak first. -> Goto 4.
- Handle next item.



Practical Agile
Agile in your Context

Why Planning poker ?

- Those who do the work estimate it.
- Emphasizes relative estimation
- Estimates are within one order of magnitude.
- Reduces anchoring - Everyone's opinion is heard.

Specification length

- Group A
  - One page spec

117 hours

- Group B
  - 7 Pages spec

173 hours

Practical Agile
Agile in your Context

Irrelevant information

- Group A

20 hours

- Group B
  - added irrelevant details:
    - End user desktop apps
    - Usernames & passwords
    - Etc.

39 hours

Practical Agile
Agile in your Context

Extra requirements

- Group A
  - Requirements 1-4

4 hours

- Group B
  - Requirements 1-5

4 hours

- Group C
  - Requirements 1-5 but told to estimate 1-4 only

8 hours

Practical Agile
Agile in your Context

Given anchor

- Group A — 456 hours
- Group B — 555 hours
  - Customer thinks 500
    - customer has no technical knowledge
    - Don’t let the customer influence you
- Group C — 99 hours
  - Same as B
    customer thinks 50

Practical Agile
Agile in your Context

Why Planning poker ?

- Those who do the work estimate it.
- Emphasizes relative estimation
- Estimates are within one order of magnitude.
- Reduces anchoring - Everyone's opinion is heard.
- Modeled for open discussion – forces thinking.
- It’s quick & fun !

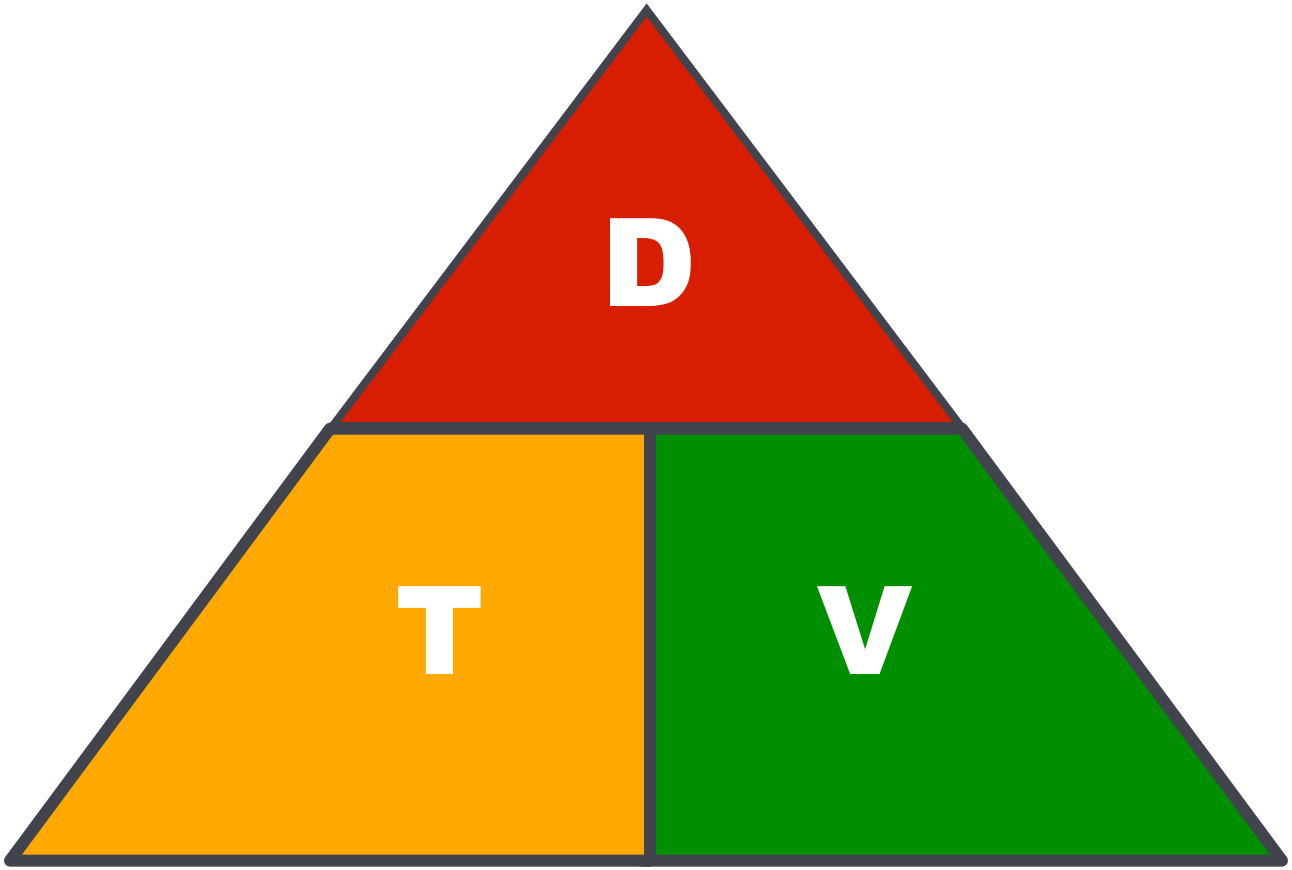
Practical Agile
Agile in your Context

Velocity

- How many points can the team complete in one iteration.

- Easy to measure.
- Fixes estimation errors.
- Easily reflects the project status.
- Primary parameter in planning.

How to calculate time ?



Practical Agile
Agile in your Context

Agile estimation and planning -

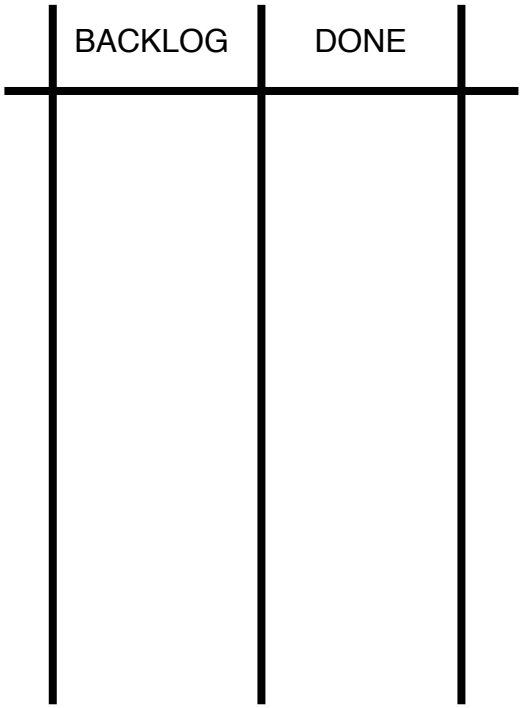
- User stories
- Splitting user stories
- Relative effort estimation
- Planning poker
- Velocity
- Release planning

Practical Agile
Agile in your Context

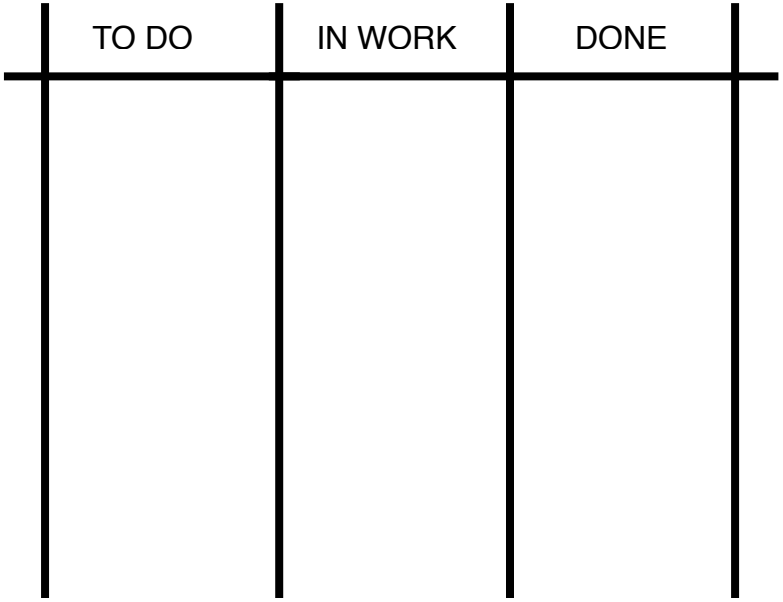

HARD
WORK
AHEAD


TEAM

Prepare your project space



DEFINITION OF DONE



Practical Agile
Agile in your Context

Create your product

- Choose a topic and a name for the product.
- Choose a product owner.
- Choose a scrum master.
- Write down 5-6 **BIG** user stories (Epics) for your product.

Practical Agile
Agile in your Context

Create your backlog

- Breakdown the Epics into smaller user stories.
  - Separate - Server \ DB \ Client.
    Recommended to color code them.
- To each user story add at least 2 acceptance criteria (Negative, Positive)
- Give **value** points to each user story.
  - Divide 100 "value points between the stories"

Practical Agile
Agile in your Context

Estimate your backlog

- Choose a story which is “average” size
  - Should be done in ~2-3 hours.
  - Break a story if you don't have one.
- Give this story 8 points (Anchor)
- Estimate the rest of the backlog using planning poker.
  - Remember to break big stories.
  - Stop when you think you have enough work for one day.

Practical Agile
Agile in your Context

Plan your project

- Prioritize the backlog using the value and effort estimate.

The next days:

- Each day will be divided to 2:
  - Theory and knowledge.
  - Exercise and practice.
    - Plan the sprint.
    - Work and stop every hour for a "daily".
      - Update board and chart.
    - Review the sprint.
      - Measure velocity.
    - Retrospect.
    - If needed:
      - Add \ Remove user stories.
      - Estimate new\updated user stories.

Practical Agile
Agile in your Context

KEEP CALM AND SEE YOU TOMORROW

KeepCalmAndPosters.com