Certified Specialist in Agile Testing (CSAT)

Syllabus

Version 1.2

Released 15-3-2022

Copyright Notice

Thank you to the main author

- Bas Kruip, Carlo van Driel

Thank you to the co-authors

- Jayapradeep Jiothis

Thank you to the review committee

- Alexis Herrera, Alfonso Fernández, Arun Janglie, Aurelio Gandarillas, Ángel Rayo Acevedo, Christine Green, Daniel Castillo Garcia, Daniel Leo Lopez Romero, Emilie Potin-Suau, Erik van Veenendaal, Fabiola Mero, Gustavo Márquez Sosa, Héctor Ruvalcaba, Isaac Marcelo Malamud Kobrinsky, Ismael Betancourt, Javier Chávez, Javier Jesús Gutiérrez Rodríguez, Jordi Fernandez, José Antonio Rodriguez, Julian Baars, Julie Gardiner,Julio Córdoba Retana, Jochem Gross, Kyle Alexander Siemens, Laksh Ranganathan, Luisa Morales Gómez-Tejedor, Márcia Araújo Coelho, Marco Fidel Peña Valbuena, Marelis V. Pérez García, Mario Alvarez Gómez, Melissa Pontes, Miaomiao Tang, Miguel Angel De León Trejo, Nadia Soledad Cavalleri, Patricia Osorio Aristizabal, Paul Mowat, Richard Seidl, Rogier Ammerlaan, Ruth Margaret Florian Caipa, Sammy Kolluru, Samuel Ouko, Sebastiaan Vreedenburgh, Sergio von Borries, Shashikumar Singh, Silvia Nane, Thomas Cagley, Tim Moore, Valeria Cocco, Wim Decoutere, Yaara Egger & Siegmond Waterfort.

Revision History

Version	Date	Remarks
0.1		Initial Beta release
0.2	30-09-2021	Revised Beta release
0.22	11-10-2021	First review and spell/grammar check
0.23	12-11-2021	Processed SIG review remarks
0.24	15-11-2021	Grammar, spelling, adding explanation team culture (par. 1.3), expansion of the definition of ‘bias’ (par. 4.1), layout improvements.
1.0	16-12-2021	Final release version
1.1	22-12-2021	Improvements on chapter 2 (PDA and overview) after internal review.
1.2	15-3-2022	Improvements on chapter 1, 4 and 6 after testruns training

Table of Content

Business Outcomes

Business objects (BOs) are a brief statement of what you are expected to have learned after the training.

BO-0	Remember all the used keywords
BO-1	Understand the principles of (personal) leadership
BO-2	Use leadership qualities to improve your team's behavior and outcomes
BO-3	How to lead the way to better quality products
BO-4	Understand the basic principles behind negotiations
BO-5	Be able to apply negotiation skills in your daily work routine
BO-6	Use different negotiation approaches in different situations
BO-7	Learn to be aware of unjudgmental probability
BO-8	Use probability in a correct way in risk assessments
BO-9	Be the Exploratory Testing expert in your team
BO-10	Practice Exploratory Testing from many different perspectives
BO-11	Understand what a bug hunt is and how it can add value to your SDLC
BO-12	Learn how to plan and perform a bug hunt
BO-13	Understand what biases are and how biases can help you become a better tester
BO-14	Be able to recognize biases by others and yourself
BO-15	Use the power of visualization
BO-16	Use a visual model to empower your improvements
BO-17	Use a visual model to empower your testing strategy and results

The positioning of a specialist in agile testing is primarily within a team, however the learned skills can be applied in a supporting role as a test coach as well. If you want to add as much value in an Agile team as possible, in the role of tester, you need the skills as described in this syllabus and as learned in the training. Obviously, the skills are also useful in a non-agile environment, however we consider the skills mandatory in an agile environment.

Learning Objectives/Cognitive Levels of Knowledge

Learning objectives (LOs) are brief statements that describe what you are expected to know after studying each chapter. The LOs are defined based on Bloom's modified taxonomy as follows:

Definitions	**K1 Remembering**	**K2 Understanding**	**K3 Applying**
Bloom's definition	Exhibit memory of previously learned material by recalling facts, terms, basic concepts, and answers.	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions, and stating main ideas.	Solve problems to new situations by applying acquired knowledge, facts, techniques and rules in a different way.
Verbs (examples)	Remember Recall Choose Define Find	Summarize Generalize Classify Understand Compare	Implement Execute Use Apply Plan

	Match Relate Select	Contrast Demonstrate Interpret Rephrase	Select

For more details of Bloom's taxonomy, please refer to **[BT1]** and **[BT2].**

Hands-on Objectives

Hands-on Objectives (HOs) are brief statements that describe what you are expected to perform or execute to understand the practical aspect of learning. The HOs are defined as follows:

- HO-0: Live view of an exercise or recorded video.
- HO-1: Guided exercise. The trainees follow the sequence of steps performed by the trainer.
- HO-2: Exercise with hints. Exercise to be solved by the trainee, utilizing hints provided by the trainer.
- HO-3: Unguided exercises without hints.

Prerequisites

Mandatory

- AU-CPAT certificate or similar certification
- Working experience in the role of tester in an agile environment

Recommended

- Having at least a 3-year working experience in agile and in testing.
- Having studied a book about agile testing, such as *Agile Testing* and *More Agile Testing* **[IN1].**

Reading instructions

This syllabus should be approached in the following manner:

- This syllabus describes the business outcomes and learning objectives for a specialist in agile testing. Please be aware that studying this syllabus alone will not allow you to pass the exam.
- In this document, the term 'tester' refers to any member in your agile team involved into testing/quality. This does not just apply to the person working in a tester's role, but also to developers, scrum masters, product owners or anybody else involved in creating a better product for your customer.

Chapter 1 – Leadership

Why do some teams perform brilliantly, and others … not so? Based on the work of Harvard's Amy Edmondson and Francis Frei, there are key elements that change a normal functioning team into a brilliant one. These key elements will be explained. Next to that, tools are presented to use this knowledge to make your team safer, more accountable, and by doing so, more productive.

Keywords

Will, Competence, Trust, Psychological safety, Value, Accountability, Responsibility, Compassion, Authenticity, Logic, Apathy zone, Anxiety zone, Comfort zone, Learning zone, PERFECT (Progress, Expectations, Role, Feedback, Effectiveness, Consequences, Tasks), LEADING (Learning culture, Effectiveness, Authority, Discussion, Ingenuity, No tolerance, Group)

LO-1.1	K1	Define the three basic elements of motivation.
LO-1.2	K2	Explain how individual motivation changes when operating in a team.
LO-1.3	K2	Explain the definition of accountability by comparing with responsibility.
LO-1.4	K2	Explain the difference between trust and psychological safety.
LO-1.5	K2	Explain what compassion, logic and authenticity mean.
LO-1.6	K2	Explain how to improve your compassion, logic and authenticity.
HO-1.1	HO-2	See how you can improve your trustworthiness and accountability.
LO-1.7	K2	Relate Apathy zone, Anxiety zone, Comfort zone and Learning zone to Psychological Safety and Accountability.
LO-1.8	K2	Explain the meaning of the PERFECT values in het context of promoting accountability.
LO-1.9	K2	Explain the meaning of the LEADING values in the context of promoting trust.
HO-1.2	HO-2	Explore how you can improve levels of psychological safety and accountability in your current team.
HO-1.3	HO-2	Execute the “My user manual” exercise.

1.1 On motivation

LO-1.1	K1	Define the three basic elements of motivation
LO-1.2	K2	Explain how individual motivation changes when operating in a team
LO-1.3	K2	Explain the definition of accountability by comparing with responsibility
LO-1.4	K2	Explain the difference between trust and psychological safety

People are driven by motivation, which is built up out of will (inner drive), competence (faith in own capabilities) and trust (willingness to be dependent on someone or something) **[LS1]**. When people operate within a team, some of the basic elements on motivation change: the need for trust (“Can I trust the road I am driving / the person next to me?”) *extends* to psychological safety (“Am I accepted for who I am?”), and the will (“I want to drive fast vehicles”) is put to use for inherent (group-)value (“I will win prices while racing for as to promote a car company I am working for”).
The best way to realize value (f.e. delivering working software that has the highest priority in the eyes of the client) is not to focus on responsibility, but accountability. Responsibility can be defined as having the intention to do the right thing right: “I’ll do my best.” Technically, it cannot be assigned

to somebody and refers to the inner will. Accountability is being liable to face consequences and it is assigned to somebody. A focus on responsibility can lead to teams performing in the so-called 'comfort zone': "we will do our best", but there is no feeling of urgency when things go wrong. When being accountable, the urgency ís present because consequences when things go wrong, are not defined by the person or team itself, but by an external (economic) agent (the client). **[LS2]**

Psychological safety can be defined as one's ability to show oneself, and behave without fear, nor feeling any negative repercussions on one's self-image, status and career. It creates a work climate where people are comfortable being themselves. As with trust, both are tight to the idea of being willing to be vulnerable, but while trust works as a cost reduction ("I don't have to be monitoring the behaviour of my surroundings"), psychological safety ("I am heard and seen and accepted") creates harmonic communication, facilitating the learning process. **[LS3]**

1.2 Improving yourself

LO-1.5	K2	Explain what compassion, logic and authenticity mean.
LO-1.6	K2	Explain how to improve your compassion, logic and authenticity
HO-1.1	HO-2	See how you can improve your trustworthiness and accountability

To be able to improve your team, start with yourself. Based on the earlier paragraph, you should focus on two concepts: trustworthiness and accountability.

Trustworthiness [LS4]

When people trust each other, they need less time and energy to focus on what the other is doing, and have more time to focus on what must be done. So, it is smart to start evaluating your own trustworthiness.
Everyone's trustworthiness is based on three elements: compassion, authenticity and logic, and people all have different/unequal strengths and weaknesses in these areas.

Compassion

A lack of compassion (that is, the ability to be able to step in one's shoes, and then helping them) is common among people who are analytical and driven. They often get impatient with those who aren't similarly motivated or who take longer than they do to understand something.
If this is your weakest link, then this is the solution: instead of focusing on what you need, radically work to ensure that everyone else gets what they need.

Logic

A lack of logic in your proposals or ideas can be the reason why people don't trust what you are saying. Solution: learn to tell a story in a good way.

Authenticity

Authenticity is, in short, showing who you are to your fellow human being. Being socially too careful or calculating drives you to do and say things what you think the other wants you to do and say, or just the opposite: you oppose to everything. In both cases psychological safety in the team is undermined. The solution? You need to flip the focus from the other, to yourself: what do *you* need, and want to say and do?

Accountability

What can you do to be an example for the team on the notion of accountability? Two strategies might help you: show more and better of what you are doing for the team (for example, make your test strategy visible on a big paper and let your team discuss this) and think about what you can do more besides executing your core activities (testing software), for example, organizing a bug hunt on a Friday afternoon.

1.3 Improving the team

LO-1.7	K2	Relate Apathy zone, Anxiety zone, Comfort zone and Learning zone to Psychological Safety and Accountability.
LO-1.8	K2	Explain the meaning of the PERFECT values in het context of promoting accountability.
LO-1.9	K2	Explain the meaning of the LEADING values in the context of promoting trust.
HO-1.2	HO-2	Explore how you can improve levels of psychological safety and accountability in your current team.

How do you improve the effectiveness of a team? Amy Edmondson **[LS2]** has investigated that question for twenty years and has identified these, earlier mentioned, two key factors: psychological safety and/or accountability.

Low levels of one and/or the other factor create cultures that produce suboptimal teams, making their members function under scary (high on accountability, low on psychological safety), unmotivating (low on accountability and psychological safety) or too relaxed (low on accountability, high on psychological safety) conditions. There are though ways to change these circumstances to the most effective culture, where teams thrive in a so called 'learning zone' (high on psychological safety and accountability).

PERFECT LEADING

The acronyms PERFECT **[LS5]** and LEADING **[LS6]** sum up these ways for improving the accountability and psychological safety of a team.

P Measure the **Progress**: a nice example is the burndown chart.
E Search for clear **Expectations**: what is the vision and mission?
R Align with the **Role:** make clear what everybody is contributing.
F Provide and ask for **Feedback**: open up to feedback and be brave enough to give it.
E Evaluate the **Effectiveness**: build in sessions on what went wrong, and why, so improvement is a standard process.
C Look at & link to **Consequences**: what will happen if the team succeeds, or fails?
T Personalize **Tasks**: make sure that every tasks has a personal owner.

L Welcome curiosity: embrace the **Learning culture**: promote the concept of 'return on experience' instead of 'return on investment': what can you 'learn by doing', instead of 'produce by delivering'?

E Put **Effectiveness** over efficiency: organizations who put the development of their staff over their output have a greater chance surviving in the long run, then those who do a cost-benefit analysis.

A Provide channels to **Authority**: find ways to push your ideas to people who have the authority to decide on it.

D Promote the **Discussion**: here critical thinking comes into play: postpone judgement while accelerating information: (f.e.) "I don't understand this idea, and it doesn't feel right for me. Can you elaborate?"

I Infuse **Ingenuity**: embrace half fabricates and don't promote perfection. Every dead end is to be seen as a way that will eventually lead to successful products.

N Apply a **No tolerance** policy: make sure that mental or emotional attacks on what people do, say, or wear, are not accepted or tolerated.

G Forge the **Group**: organize teambuilding-events, eat together, use the "This is my manual" template. People who know each other better, have a greater chance of trusting and accepting each other.

1.4 Learning to know one another

HO-1.3	HO-2	Execute the "My user manual" exercise

The "My user manual" exercise **[LS7]** is a fast and low level means to get a grip on the elements that motivates and demotivates individual members of the team. The overview will make it easy to be able to contribute to a more psychological safe culture: if every member is approached with this manual, one feels more acknowledged. The following parts are expressed per team member:

1. A picture or drawing of the student as a person
2. Desired work conditions
3. Desired work hours / times
4. Desired communications method
5. Desired feedback method
6. Things needed to be able to perform well
7. Personal pitfalls
8. Things that the student loves at work
9. Other things that team members need to know about you

Chapter 2 – Damage

At the heart of testing there is often no clear idea of what risk is. And there are some problems with the concept, which will be explained. As a result, the better concept "Damage" will be introduced. Damages 'live' in 4 different categories, and need different approaches to prevent them, or mitigate their impact. How this is done is also shown.

Keywords

Knowns, Unknowns, Damage, Naive constructivism, Potential Damage Analysis (Gathering, Prioritise, Bowtie, Follow up), Bowtie elements (Threats, Consequences, Controls, Mitigation, Prevention, Escalation), P*I, Knowledge, Black swans, High Reliable Organization (HRO), BARE (Big picture, Accountability and psychological safety, Redundancy, Evolve aggressively), SPACE (Situational awareness, Preoccupied with failure, Add enough depth, Commit to resilience, Esteem for experts), Feedback (Recognize, Create, Model), PDCA (Plan Do Check Act), Management by Objectives (MbO), Management by Learning (MbL).

LO-2.1	K2	Understand the concept of 'naive constructivism'.
LO-2.2	K2	Compare the concept 'damage' to that of 'risk'.
LO-2.3	K2	Understand the working of the bowtie.
LO-2.4	K2	Understand how to get a clear view on priorities while using P*I.
LO-2.5	K2	Understand how Knowledge is of influence on the quality of a PDA.
LO-2.6	K2	Understand how to deal with black swans.
LO.2-7	K2	Understand the importance of the Follow up.
LO-2.8	K2	Understand every element of the acronym BARE SPACE.
LO-2.9	K2	Understand how BARE SPACE can be implemented in an agile culture.
LO-2.10	K2	Understand how feedback helps to get more insight on unknown (potential) damages.
LO-2.11	K2	Understand how seeing, playing and modelling create feedback.
LO-2.12	K2	Understand the concepts Management by Objectives (MbO) and Management by Learning (MbL).
HO-2.1	H-02	Execute a PDA.

2.1 Introducing damage

LO-2.1	K2	Understand the concept of 'naive constructivism'.
LO-2.2	K2	Compare the concept 'damage' to that of 'risk'.

Risk is at the heart of testing, but there is often no clear idea of what it really is. More importantly, the concept of 'risk' is a potential bias that can lead to unforeseen disasters. How? Risk is automatically seen as an *attribute* of a product (or service). But a product can, by itself, be a hazard: a web shop is a new means to sell goods, but is also a new way to lose goods, goodwill and clients. This 'naive constructivism' must be rooted out as a mindset, and for that reason the concept "damage" is introduced. Damage is defined as "any negative value of a product that matters to a person that matters." While containing risk ("anything that threatens the value of a product that matters for a person that matters"), 'damage' sets a broader, and more complete scope.

Known and unknown damage

As covered in the CPAT, things, like damage, can be known, or unknown. While known threats can be taken care of with preventing and / or mitigating measures, unknown threats ask for a different approach: exploration ("What possible ways to use this web shop can lead to disaster?") and vigilance ("Something is going wrong, but we don't know what.").

Damage and potential damage

Next to the known and unknown category, known damage can have occurred (Sales drop because of a big bug so clients can no longer shop) or can potentially occur (The web shop is built on an old hardware platform, and support on that hardware will stop soon). This also applies to unknown damage: it happens (a believed popular product is never ordered. Something is wrong…) or can happen.

Based on those two variables, damage can be placed in four categories:

- Known damage
- Known potential damage
- Unknown damage
- Unknown potential damage

For known damage, testers have a thing in place, called a bug report. But what about the other categories?

2.2 Known Potential Damage Analysis (PDA)

LO-2.3	K2	Understand the working of the bowtie.
LO-2.4	K2	Understand how to get a clear view on priorities while using P*I.
LO-2.5	K2	Understand how Knowledge is of influence on the quality of a PDA.
LO-2.6	K2	Understand how to deal with black swans.
LO.2.7	K2	Understand the importance of the Follow up.
HO-2.1	H-02	Execute a PDA.

A PDA (potential damage analysis) consists of four steps, described below.

1. **Gathering potential damages**

With the use of a brainstorm session (in any form), all potential damage concerning new or changed circumstances (project or product) need to be listed.

2. **First things first**

When all potential damages one can come up with are defined, prioritizing can begin. These potential damages are arranged by prioritizing them by using the P (Probability) * I (Impact) calculation.

Figure 2.2

		CONSEQUENCES – WHAT IS THE MAXIMUM REASONABLE CONSEQUENCE				
		Insignificant	Minor	Moderate	Major	Catastrophic
LIKELIHOOD RATING	Almost certain	Medium	Medium	High	Extreme	Extreme
	Likely	Low	Medium	Medium	High	Extreme
	Possible	Low	Low	Medium	High	High
	Unlikely	Low	Low	Low	Medium	High
	Rare	Low	Low	Low	Low	Medium

Knowledge

An often-ignored aspect of prioritizing [**PD2**], is that the stakeholders who prioritize, might not have enough knowledge on the matter to create a prioritarization of enough quality. The question that always should be asked is: do the people involved in this prioritization have enough K (Knowledge) in the matter to assure the quality of the outcome?

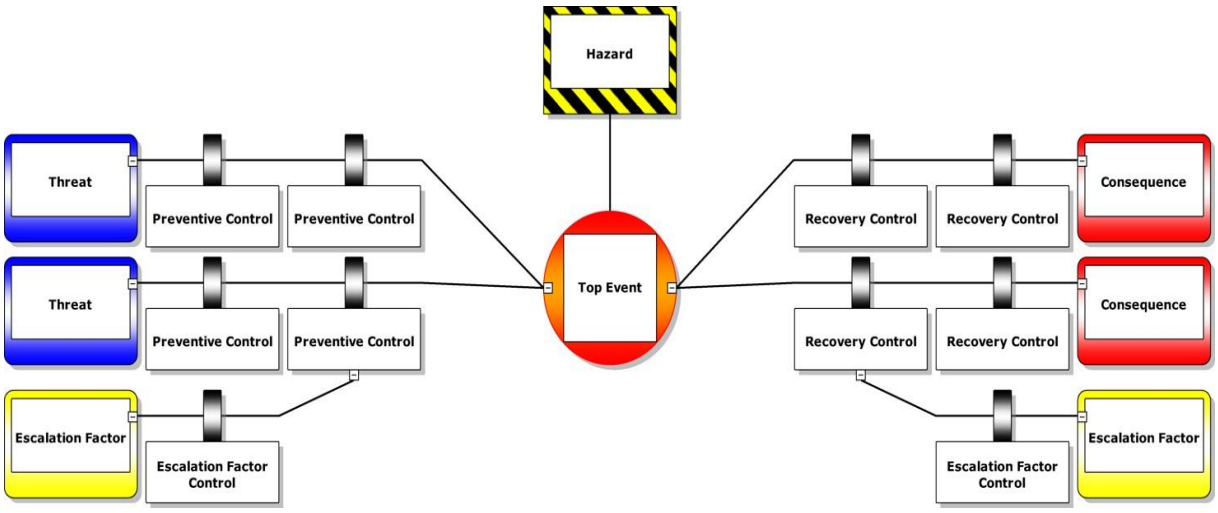
Black Swans

Black swans [**PD2**] are potential damages that people don't imagine, or dó imagine (but based on the assessment that the chances of those damages are considered too low, do not follow up on). A way to get a clear sight on black swans, is to organize a bowtie and prioritization session with general experts who are not part of the project, ánd are highly critical about what is being made.

3. Executing a Bowtie

The bowtie method [**PD1**], well known in general quality management, helps the team to brainstorm over a particular potential damage, ánd forces its members to come up with solutions.
It is a means to get a clear view on threats and consequences when things are put to use, as well as on preventive and mitigating measures. It gets its name 'bowtie' because of the bowtie shape of the tool. On the one side it depicts threats and measures to prevent the threat from happening (preventive controls), on the other side it depicts consequences might the damage still occur, and its mitigating measures (recovery controls). Escalation factors (new damage that can occur because of the preventive or mitigating measures) ask in their turn for new preventative and / or mitigating measures (escalation factor controls).

Figure 2.1



The client has prioritized a "search engine" as an important feature for the next sprint. The hazard is defined as "using the search engine" and a top event is defined as "search engine does not show the wanted article."
The team comes up with the following threats: "function does not pick up certain characters", and "function does not show items out of stock".
The preventative measures the team comes up with are: "shopper gets warning when using illegal characters" and "paste function not allowed in search field", respectively "alternative articles will be shown when searched article is out of stock."
Also, consequences are defined: "function shows no articles" and "function shows too many other articles".
Mitigating measures are these: "shopper gets message with apologies and suggestions on synonyms that occur in database" and "filter options". An escalation factor can then be that too many filter options are shown. An escalation factor control is defined as "shopper must preselect a product category before filter options are shown."

Finally, the bowtie method allows one to see the combinations of threats and consequences for insight purposes:

"Function does not show items out of stock, therefor function shows no, or too many articles".

4. Take care of the Follow up

After priorities have been set, preventative and mitigating measures are created. Often enough these measures are secured in user stories, but what if they are, or cannot?

"A threat that was found during the bowtie session, was the absence of a UX-specialist in the team. The potential damage was considered high, but no tester or developer would pick it up."

All potential damage that is considered problematic enough, must have a follow through. For every such a potential damage, an accountable person (who), a timeline (when) and the how must be defined.

2.3. Dealing with unknown damage

LO-2.8	K2	Understand every element of the acronym BARE SPACE.
LO-2.9	K2	Understand how BARE SPACE can be implemented in an agile culture.
LO-2.11	K2	Understand how feedback helps to get more insight on unknown (potential) damages.
LO-2.11	K2	Understand how seeing, playing and modelling create feedback.

When dealing with unknown and potential unknown damage, High Reliable Organizations (HROs) [**PD2**], like parts of the military, flight control and firefighters, are experts. They cannot lay back and pick up the pieces afterwards, because the consequences would damage occur, are often too high a price to pay. What can people learn from HRO's if it comes to unknown (potential) damages?

Unknown potential damages

HRO's have developed a few strategies to deal with unknown potential disasters. The main goal of these strategies (the acronym is for them is "BARE") is to push for excellent teams:

1. Continuously communicate the **B**ig picture (mission, vision and how the teams fit in)
2. Assure high standards of **A**ccountability and psychological safety (see chapter 1, Leadership) and adjust accordingly.
3. Organize **R**edundancy, preventing any organization (like a team) to stop, when parts fail.
4. **E**volve aggressively: seek to know what is not known and create an informed culture.

Unknown damages

Unknown damages are damages that occur, but where nobody involved is consciously aware of.

"I don't understand. I have bought 100 pairs of very popular shoes for our web shop, but nobody will buy them. Oh… whatever, these things always just work out fine."

HRO's have a set of rules (the acronym for them is "SPACE"), that keeps their members constantly on high alert:

1. **S**ituational awareness: people have an intuition, that is build up through experience. The rule is to always act on it.
2. **P**reoccupation with failure: all members should be at a constant *unease*: "What did we miss?"
3. **A**dd enough depth: noticed irregularities should trigger critical thinking (for more information, see the CPAT). Easy assumptions are *always* inadequate.
4. **C**ommit to resilience: train mind, body and heart to take a blow and recuperate fast.
5. **E**steem for Expert: change the command structure when the situation calls for it: a team leader on the floor is more equipped to assess an unforeseen and critical situation and take action, then a general manager is.

What can this do for an agile tester?

Some strategies and ruling are already in some way implemented in an agile tester's way of working: you can for example look at exploratory testing as an implementation of the strategy "evolve aggressively" and the rule "situational awareness". In a broader sense, "commitment to resilience" is incorporated in an agile team with one of the principles of agile: "Accommodate change".

But why not make “situational awareness” or “preoccupation with failure” a simple rule? And how should one organize the strategy “Redundancy” in a team or project?

Feedback

Closing the information gap (that is, pulling unknown potential damages into the “known” area) needs a learning culture. A learning culture can be described as a culture where the Plan Do Check Act cycle is produced as often and fast as possible as to create as much feedback as possible.

See the feedback

You’ll be aware of this cycle when you understand that the known elements in your day to day working life all inform you: refinement sessions, automated pipelines, evaluations and stand-ups, they all provide feedback which is the information you gain when entering the “Check”-step of the PDCA cycle.

Play for feedback

Another way to learn is to actively create feedback. You can do this by, for example creating popcorn flows (see chapter 7.3.1.) or use a technique called ‘system thinking’, which asks to look at the same object of interest from different perspectives: what relations does it have? What parts are there? What does it produce? Does it have omissions or unclarities? What is its relationship with a concept like “time”?

Model for feedback

Make what you think visible, tangible. Create a model of the object of interest: a technical drawing of all databases involved, or all functions and their interactions. Write down your model of the project or product (see for more information on this one the CPAT) and hang it on the wall.

2.4. Overview

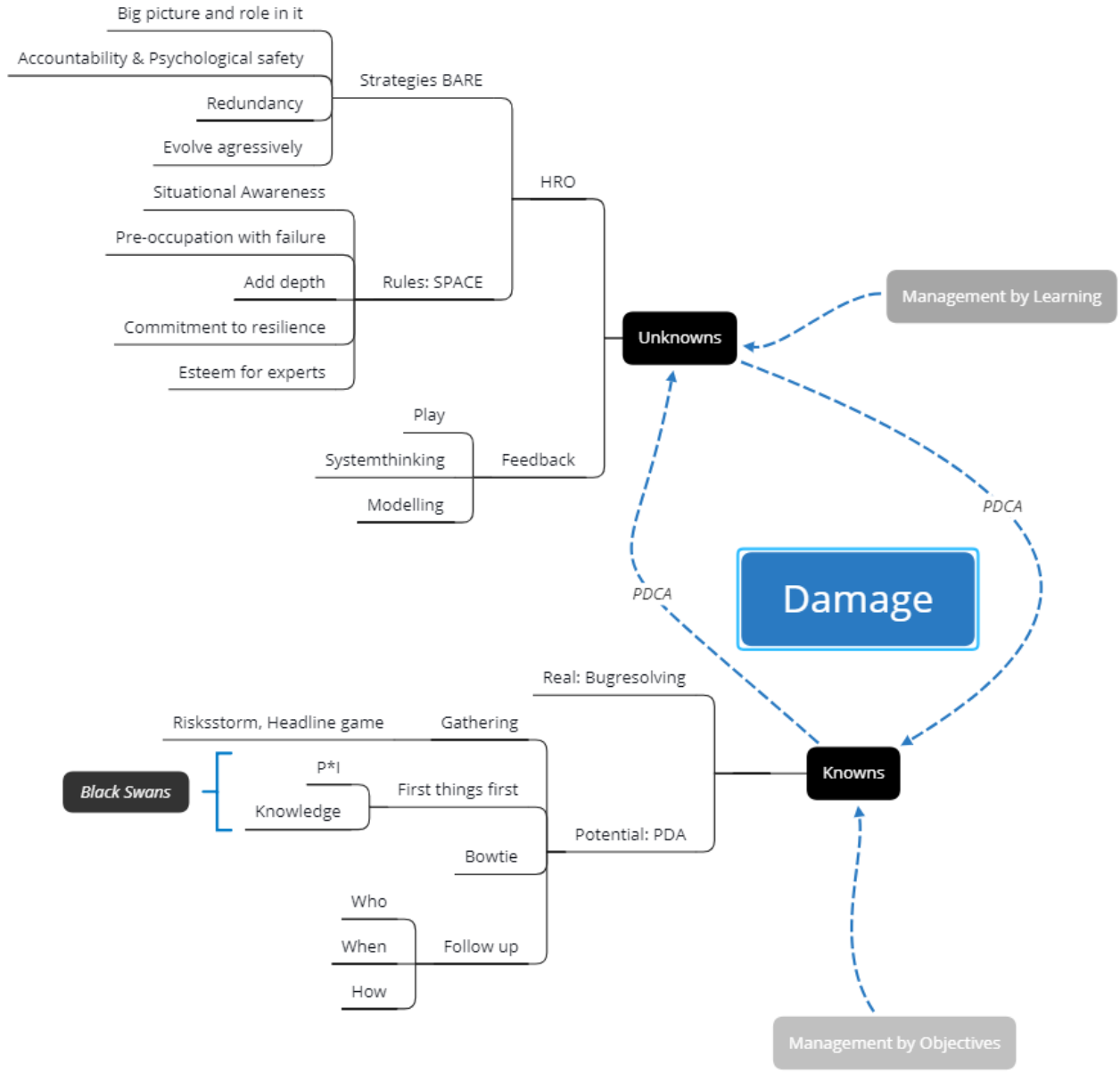
LO-2.12	K2	Understand the concepts Management by Objectives (MbO) and Management by Learning (MbL)

In the picture below, all the elements concerning (potential) damages that were mentioned earlier are depicted in a grand overview. Two last elements are added:

Management by Objectives (MbO) & Management by Learning (MbL)

When damages are known, one should promote a 'Management by Objectives' culture: the problems are clear, and measures are to be build (objectives need to be reached). When looking for, or anticipating on damages that are yet unknown, a culture of 'Management by Learning' should occur: facilitate a culture where people can learn (developing skills and acquiring knowledge).

Figure 2.3



Chapter 3 – Bug hunt

A bug hunt **[BH1]** means exactly what the word already suggests: a hunt for bugs. A bug hunt can however create a lot of value in a very short period and is a very good add-on to all the so-called 'regular' tests.

Keywords

Bug hunt, test charters, soap opera scenario, feedback

LO-3.1	K2	Define the perspective of a bug hunt
LO-3.2	K2	Be able to demonstrate the value of a bug hunt
LO-3.3	K2	Learn when to organize a bug hunt
LO-3.4	K2	Be able to explain how a bug hunt is performed
LO-3.5	K2	Be able to demonstrate the value of soap opera scenarios
LO-3.6	K2	Relate soap opera scenarios to bug hunting
HO-3.1	HO-2	Execute a bug hunt
HO-3.2	HO-2	Plan a bug hunt
LO-3.7	K2	Explain the value and risks of bug hunts
LO-3.8	K2	Be able to be a bug hunt ambassador in your company

3.1 Definition and perspective of a bug hunt

LO-3.1	K2	Define the perspective of a bug hunt
LO-3.2	K2	Be able to demonstrate the value of a bug hunt
LO-3.3	K2	Learn when to organize a bug hunt

A bug hunt is an event in which a group of people sit together and hunt for bugs in a specific application. 'Hunting' is considered to actively search for unknown problems in the application with a group a people. The goal is to deliver as much new insights as possible in a short period of time.

The perspective of a bug hunt is to:

- have a look at the product from a fresh angle
- break daily routine

However, a bug hunt does not replace your regular testing, as tests done in a bug hunt are usually more superficial and complex situations and interfaces are not covered.

Bug hunts can be organized for many reasons and at many moments. For example, before every release, when you have too many production incidents, or every week, every month, when you have lost focus, as team building or as knowledge sharing event.

The bug hunt should be part of your test strategy and should be performed regularly, but not too often, as it can lose its effect:

- chances of finding new bugs decrease
- it takes more time for the attendees if done very regularly

- it becomes ‘work’ instead of ‘fun’

Bug hunts losing effect can be avoided by rotating the people involved in the bug hunts. Do not use the same participants all the time, make different groups that rotate in your bug hunt cycle, for example:

- week 1: group 1: people from finance and the application managers
- week 3: group 2: HR and key-users
- week 5: group 3: managers and end-users
- week 7: start over with group 1

3.2 How does a bug hunt work?

LO-3.4	K2	Be able to explain how a bug hunt is performed
LO-3.5	K2	Be able to demonstrate the value of soap opera scenarios
LO-3.6	K2	Relate soap opera scenarios to bug hunting
HO-3.1	HO-2	Execute a bug hunt
HO-3.2	HO-2	Plan a bug hunt

The ingredients of a bug hunt are:

- groups of 2-3 people
- a moderator
- optional pre-defined test charters (defined by the tester)
- hotel bells
- a judge
- optional bug categories (see 3.3)
- prizes.

The bug hunt process is:

- create the groups
- give each group a bell to use when a bug is found to create a bit of competition between the different groups
- explain the rules:
    - about the timeslot (45 mins to 2 hours)
    - how to use the bell
    - how to register bugs
- optionally let them select a pre-defined test charter
- show the prizes (f.e.: sweets, fruits, free lunch) for extra motivation
- start testing (no rules on how people test)
- after the timeslot has passed:
    - give the prizes to the teams that won in each bug category.
    - do not forget to share knowledge among the people involved, about the product and how bugs were found and dealt with.

Besides pre-defined test charters to ‘steer’ the direction of a bug hunt, it is also an option to do the bug hunt without any pre-defined format. In that case, it can be helpful to explain soap opera

scenarios **[BH2]** as these may help finding bugs more easily. Soap opera scenarios are test scenarios created from and inspired by the real world. They differ from regular test scenario's by following the TV-soap scripts: compact a series of events that usually take years into 30 minutes: get married, get sick, get pregnant, die during labor when an airplane crashes into the hospital. The idea behind soap opera scenarios is that if your system can manage these extreme scenarios, it can also manage normal behavior.

Soap opera scenarios demand test knowledge and business knowledge and are a great way to share this knowledge among development teams and business by testing together and doing a retrospective. They have a broad coverage of the system and do not need system requirements; they challenge your creativity and often help find design holes in the system.

3.3 The value and risks of bug hunts

LO-3.7	K2	Explain the value and risks of bug hunts
LO-3.8	K2	Be able to be a bug hunt ambassador in your company

Besides to deliver as much new insights as possible in a short period of time, bug hunts can provide other value to your team and to the system as well:

- engage (key) users that are less involved in developing the product
- make people enthusiastic about the product
- people share knowledge & learn about the business and the product
- create team building
- create an accessible introduction to exploratory testing
- have fun

Bug hunts also have some risks:

- no review is done at the end; people have not learned from each other
- you run the risk of giving your team and system bad publicity if the found bugs are basic scenario's that should work, so be sure that some other testing is done before doing a bug hunt. This keeps the bug hunt effective and the users motivated.

Make bug hunts part of your usual test strategy and perform them regularly with different groups of people. You can even decide to do a large bug hunt by performing crowd testing, also called crowdsources testing **[BH3]**.

A general recommendation to a bug hunt is to think carefully about your bug categories. Do not let competition take over by creating a category 'most bugs'. Such a bug category will not give you the best result, with the most valuable bugs. Instead, use categories such as 'strangest bug, 'most complex to reproduce bug' or the 'why would anyone do that bug?'.

Finally, you may have to do some ambassador work in your company to ensure that bug hunts are generally accepted. In order to implement it, start by being convinced of the value of bug hunts yourself, by not asking for permission to organize them (better ask for forgiveness if needed) and by removing as many obstacles as possible. One obstacle could be time, so organize a bug hunt during a lunch break, at a good location, and do not ask people to do preparations. Arrange the lunch and ask managers to join as well, as, in the end, they will save money and time thanks to bug hunts.

Chapter 4 – Biases

"Without the aid of prejudice and custom, I should not be able to find my way across the room" and "Prejudice is the child of ignorance" **[BI1].** Without our biases, our life would be impossible to live. However, biases also trick us without us noticing, or people can use our biases to their advantage.

Keywords

Framing bias, Anchoring bias, Halo effect bias, Apophenia bias, Conflict of Interest bias, Attribution bias, Confirmation bias, Status-Quo bias, Prejudice, Ignorance, Judgment, Cognitive limitations, Social conformism, Beliefs, Conflicts of interest, Statistical biases

LO-4.1	K2	Classify the reasons for having biases
LO-4.2	K2	Explain why biases are a good and a bad thing
HO-4.1	HO-3	Practice with a prejudice bias
HO-4.2	HO-3	Practice with a cognitive bias
HO-4.3	HO-2	Experience the Anchoring bias by demonstrating it in the group
LO-4.3	K3	Apply the Anchoring bias to testing
LO-4.4	K3	Apply the Framing bias to testing
LO-4.5	K3	Apply the Halo & Horn effect bias to testing
HO-4.4	HO-2	Experience the Halo & Horn effect bias by demonstrating it in the group
LO-4.6	K3	Apply the Statistical bias to testing
LO-4.7	K3	Apply the Apophenia bias to testing
LO-4.8	K3	Apply the Conflicts of interest bias to testing
HO-4.5	HO-2	Experience the Attribution bias by demonstrating it in the group
LO-4.9	K3	Apply the Attribution bias to testing
HO-4.6	HO-2	Experience the Dunning-Kruger effect by demonstrating it in the group
LO-4.10	K3	Apply the Dunning-Kruger effect to testing
HO-4.7	HO-2	Experience the Confirmation bias by demonstrating it in the group
LO-4.11	K3	Apply the Confirmation bias to testing
LO-4.12	K3	Apply the Status-quo bias to testing
HO-4.8	HO-2	Experience the Status-quo bias by demonstrating it in the group
HO-4.9	HO-3	Experience your own prejudices by making a Harvard test
LO-4.13	K2	Summarize ways to overcome your biases

4.1 The What and Why of biases

LO-4.1	K2	Classify the reasons for having biases
LO-4.2	K2	Explain why biases are a good and a bad thing
HO-4.1	HO-3	Practice with a prejudice bias
HO-4.2	HO-3	Practice with a cognitive bias

The word 'bias' has many definitions. The following definition from the Merriam Webster Unabridged Dictionary **[BI2]** is used:

- An inclination of temperament or outlook
    - Especially: a personal and sometimes unreasoned judgment: PREJUDICE

- An instance of such prejudice
- Bent, Tendency
- The systematic error introduced into sampling or testing by selecting or encouraging one outcome or answer over others

All people are biased, whether they like it or not. This can be explained by 4 reasons:

- Cognitive limitations; people have limited brain capacity.
- Social conformism; humans are influenced by social pressure and others. We like to blend in.
- Personality and beliefs; people all have their own set of beliefs.
- Noise, communication problems; people tend to listen to noise and it is hard to communicate clearly.

The biases are divided into 2 groups/classifications:

- Biases that are connected to redefining your external reality; biases that are occurring outside your own internal world of belief. F.e.: the fly in the urinal to reduce cleaning costs. This happens in your external reality; it is not your own believe.
- Biases that are connected to redefining your internal reality; biases that are occurring within your own internal world of belief. F.e.: you have a prejudice against woman. Your internal reality is telling you that woman cannot drive a car.

Biases are not always a bad thing. According to our mental models (AU-CPAT training) and D. Kahneman **[BI3],** system 1 thinking people could not function in a 'normal' way without biases. Without biases people would have to use system 2 for everything they do and that is not possible due to their limited brain capacity.

4.2 External reality biases

HO-4.3	HO-2	Experience the Anchoring bias by demonstrating it in the group
LO-4.3	K3	Apply the Anchoring bias to testing
LO-4.4	K3	Apply the Framing bias to testing
LO-4.5	K3	Apply the Halo & Horn effect bias to testing
HO-4.4	HO-2	Experience the Halo & Horn effect bias by demonstrating it in the group
LO-4.6	K3	Apply the Statistical bias to testing
LO-4.7	K3	Apply the Apophenia bias to testing
LO-4.8	K3	Applying the Conflicts of interest bias to testing

Being aware of different types of biases will help to recognize them and to overcome them or to put them to good use for your testing.

4.2.1 Anchoring bias

Anchoring is to rely on the first piece of information encountered when making decisions **[BI4]**.

According to this heuristic, individuals begin with an implicitly suggested reference point (the "anchor") and make adjustments to it to reach their estimate. F.e. ask people how many sweets are in a can by asking: are there more or less than 100 sweets in the can? People will give answers that are around the 100 mark. The same question with 200 sweets will give answers around the 200 mark.

Testers need to take into consideration when testing if the given information is correct. No assumptions are to be made about the numbers and you could have been anchored and thus performing the wrong tests:

- How many users are expected? Answer: 3000, however in real life only 500). You probably start your test around 3000.
- How fast should the system be? Answer: <0,3 sec, however in real life users accept <1 sec. You start raising bugs on the performance and developer use a lot of time to make things faster without adding any value.
- How many bugs were there in the previous year? Answer: 5, however in real life: 500. You probably think the software is pretty good and well tested.
- Base rate fallacy: people give one piece of information more value than other pieces, which results in false calculations. F.e. Steve is shy. Is Steve A) a salesperson b) a librarian. Answer: salespersons are not shy, so he must be a librarian. However, there are >100000 times more salespeople than librarians, so the chance of Steve being a salesperson is way bigger. We have forgotten to use the total amount of people per job in our decision making.

4.2.2 Framing bias

Framing influences the way people organize, perceive, and communicate about reality. It can be positive or negative, depending on the audience and on what kind of information is being presented. F.e. you know you talk very fast and most people have a problem with that. At the start of an interview you place the next frame: "when I get excited about a project I tend to start talking very fast, so please slow me down when that happens". As soon as you start talking very fast in the minds of the interviewers you are very excited about the project. They no longer irritate themselves when you speak too fast, the frame helped to change it into something positive.

In social theory, 'framing' is a schema of interpretation, a collection of anecdotes and stereotypes, that individuals rely on to understand and respond to events. People use filters to make sense of the world, the choices they then make are influenced by their creation of a frame **[BI5]**.

In testing, framing is sometimes used when stakeholders or management want to speed up deliveries or get their piece of functionality in a sprint, or when a company decides to implement a test tool, not because it is the best tool for the team, but for other reasons. Framing does not have to be negative; it can also be used in a positive way if you want to make changes to improve in spite of unwilling people. Framing on changes can help implement these changes.

4.2.3 Halo and Horn effect

The halo effect and the horn effect happen when an observer's overall impression of a person influences their feelings about that person **[BI6]**.

The name 'halo effect' is based on the concept of Saint's haloes, and is a specific type of confirmation bias, wherein positive sentiments in one area cause questionable or unknown characteristics to be seen positively. If the observer likes one aspect of something, they will have a positive predisposition toward everything about it.

The opposite of the halo is the horn effect, when individuals believe that (negative) traits are inter-connected. The term 'horn effect' refers to Devil's horns. It puts a negative twist on perceptions: if the observer dislikes one aspect of something, they will have a negative predisposition towards other aspects.

An example of the halo effect in testing is that testers tend to have this bias towards developers; this developer looks nerdy, he is probably a good developer (and has some form of autism and is difficult to talk to and …)

An example of the horn effect in testing is: you meet that new tester who is from India and you have worked with another tester from India in the past who was very bad in testing and you immediately decide that this new tester is a bad tester as well.

4.2.4 Statistical bias

Statistical bias is a systematic tendency in the process of data collection, which results in lopsided, misleading results **[BI7]**.

It is a property of a statistical technique or of its results whereby the expected value of the results differs from the true underlying quantitative parameter being estimated. This can occur in the way the sample is selected, or in the way data are collected.

In testing, for example, test data is being used and often selected from a source. Is this data on which testers decide to test (or not) correct or are numbers selected in an undesired way?

An example: a tester needs test data for a health insurance company to test the newly build calculator for the monthly fee in relation to the expected profit. The test data only consist of you people between 10 and 25 of age. They have a low healthcare cost profile, so probably you will get a very high profit. This is however not a representation of the total set of customers of that health insurance company.

4.2.5 Apophenia bias

Apophenia, also known as patternicity, is the human tendency to perceive meaningful patterns within random data **[BI8]**.

Apophenia is well documented as a rationalization for gambling. Gamblers may imagine that they see patterns in the numbers which appear in lotteries, card games, or roulette wheels. Instead of doing proper risk/change calculations, people still believe that after 10 times black, the chance of red will be bigger than the chance on black. In reality, the changes are 50/50 each time you play.

In testing, for example, testers tend to use patterns they think they know and estimate risks based on that. This 'gut feeling' calculation is often wrong. Testers should do proper calculation based on the numbers. Testers may use certain numbers or sequences of numbers all the time because they have caused issues in certain applications in the past. However, they fail on checking if the problem actually has anything to do with that specific number(s). Many people try to derive meaning out of a repetition of numbers where none exists.

4.2.6 Conflict of interest

A conflict of interest is a set of circumstances that creates a risk that professional judgment or actions regarding a primary interest will be unduly influenced by a secondary interest **[BI9]**.

The potential conflict is autonomous of actual improper actions, it can be found and intentionally defused before corruption, or the appearance of corruption, happens.

In testing, for example, this can happen when prioritizing the backlog or risks. Stories and risks can get a higher or lower value based on interests or external influences instead of real value. The security risk can be very high because the director just saw on the news that another company was hacked and he forces the team and tester to run a full security check, even if there is no risk.

4.3 Internal reality biases

HO-4.5	HO-2	Experience the Attribution bias by demonstrating it in the group
LO-4.9	K3	Apply the Attribution bias to testing
HO-4.6	HO-2	Experience the Dunning-Kruger effect by demonstrating it in the group
LO-4.10	K3	Apply the Dunning-Kruger effect to testing
HO-4.7	HO-2	Experience the Confirmation bias by demonstrating it in the group
LO-4.11	K3	Apply the Confirmation bias to testing
LO-4.12	K3	Apply the Status-quo bias to testing
HO-4.8	HO-2	Experience the Status-quo bias by demonstrating it in the group
HO-4.9	HO-3	Experience your own prejudices by making a Harvard test

4.3.1 Attribution bias

In psychology, an attribution bias is a cognitive bias that refers to the systematic errors made when people evaluate or try to find reasons for their own and others' behaviors **[BI10]**.

The most famous attribution bias is the self-serving bias: this is any cognitive or perceptual process that is distorted by the need to maintain and enhance self-esteem, or the tendency to perceive oneself in an overly favorable manner. It is the belief that leads individuals to ascribe success to their own abilities and efforts, but to ascribe failure to external factors. When individuals reject the validity of negative feedback, focus on their strengths and achievements but overlook their faults and failures, or take more credit for their group's work than they give other members, they are protecting their ego from threat and injury.

An example related to testing is:

- Think of a specific time when a developer spoke the famous words: “it works on my machine”.
    - What happens?
    - Why does that developer say that?
    - ➔ Invariably, you’ll get a mix of situational (“I was in a hurry”, “pressure to deliver”) and dispositional (“jerk” or “autistic guy”) attributions.
- Next, think of a specific time when you, as a tester, were rude to a developer and the reason why you were rude.
    - ➔ You probably give 100% situational attributions (for example: “pressure to deliver”, “error in test environment” or “in a hurry”).

There is a bias somewhere as it is impossible for both ratios of situational/dispositional attributions to be correct simultaneously for all people.

4.3.2 Dunning-Kruger effect

The Dunning–Kruger effect **[B15]** is based on the idea that for example people who start a new job tend to overestimate their abilities and performance, because they lack knowledge and experience concerning the baseline of performance. On a broader scale, this effect directs to the idea that people have problems to assess their own performance in a neutral way from a second person view: we have the tendency to think that we know what we are doing, and base that idea on a good, gut feeling, while from a detached point of view, this is very possibly not the case.

In testing, this bias is a reality when calculating what the chances are that some sort of damage might occur: teammembers often overestimate their knowledge and their for correct asessment of a possible threat.

4.3.3 Confirmation bias

Confirmation bias is the tendency to search for, interpret, favor, and recall information in a way that confirms or supports one’s prior beliefs or values **[BI11]**.

In testing, people are confronted with this bias a lot. Testers have to stay focused on the confirmation bias as it will stop them from really exploring an application for the ‘unknowns’. It can be as easy as hanging up a sticky note with ‘confirmation bias’ on the screen or putting it in the template for the session-based tests.

4.3.4 Status quo bias

Status quo bias is a preference for the current state of affairs. The current baseline (or status quo) is taken as a reference point, and any change from that baseline is perceived as a loss **[BI12]**.

In testing, for example, testers keep on running the same tests and way of testing because they are familiar with it? Or do they try new things and get out of their comfort zone to find new things and learn?

4.3.5 Prejudices

Prejudice is prejudgment or forming an opinion before becoming aware of the relevant facts of a case **[BI13]**.

The word is often used to refer to preconceived, usually unfavorable, judgments towards people or a person because of their gender, political opinion, social class, age, disability, religion, sexuality, race/ethnicity, language, nationality, or other personal characteristics.

This bias is not specifically related to testing. Everybody has prejudices, whether you like it or not. To get the confirmation of your prejudices take a look at https://implicit.harvard.edu/implicit/takeatest.html

4.4 How to overcome biases

LO-4.12	K2	Summarize ways to overcome your biases

In order to overcome biases, people can gain awareness by:

- Implicit Associations Test:
  - The first step to changing one's implicit biases is acknowledging that they have them. One can check their level of implicit bias by taking one (or several) of the Implicit Association Tests (https://implicit.harvard.edu/implicit/takeatest.html).
- HALT (**H**ungry, **A**ngry, **L**onely, **T**ired) **[BI14]**
  - Make no critical decisions if one is **H**ungry, **A**ngry, **L**onely, or **T**ired. People's system 1 will take the decision and it will very likely be heavily influenced by one's biases.
- Relational impact
  - Consider who is impacted by your decision (or lack of decision). Sometimes, looking at how others will be impacted by a given decision will help you clarify the decision you will take.
- Rational Analysis
  - Recall that many of the biases work very quickly and rely on intuition. Although intuition is a valid part of your decision-making process, you should check whether there are any actual observations you can make or data for this decision.
- Outsider Perspective
  - Sometimes, decisions are difficult to make because people don't have enough data and/or experience. Sometimes, decisions are difficult to make because people have conflicting values and priorities for the outcomes of a decision. This is precisely where it is valuable to ask a reliable source for input. They might possess the data or experience you lack.
- Reflect on the past
  - Look back on your decision-making history and ask if you have ever been in a situation like this before. How was that situation similar to the current one? How was it different? What were the outcomes? How did you make that decision in the past and what influenced your choice?

Chapter 5 – Exploratory Testing

It is always easy to miss something you are not looking for. By sticking to the 'knowns', such as specifications, you limit your insights about the software. A way to overcome this is to search for the 'unknowns' by using Exploratory Testing **[ET1]**.

Keywords

Models, Focus, De-focus, Knowns, Unknowns, Dimensions

LO-5.1	K2	Summarize the general dimensions of Exploratory Testing
LO-5.2	K2	Summarize the sequences and interactions dimensions of Exploratory Testing
LO-5.3	K2	Summarize the entities and relationships dimensions of Exploratory Testing
LO-5.4	K2	Summarize the states and transitions dimensions of Exploratory Testing
LO-5.5	K2	Summarize the ecosystem dimensions of Exploratory Testing
HO-5.1	HO-2	Create ET test charters using the different dimensions of Exploratory Testing
HO-5.2	HO-2	Execute the created test charters to reveal new insights

5.1 Models

Models are a simplification of the real world. People create and use conceptual models for the software they design and build, and they have mental models in their head which they use to test the actual software (in the real world).

The more people connect to their mental models, the more they go into a focused mode. This can be valuable to detect potential issues in a detailed part of the software. People, however, have to de-focus regularly not to lose sight of the big picture, and to be able to estimate the value they are working on.

5.2 Positioning of Exploratory Testing

There are things that we know that we know (facts), things that we know we do not know (we ask questions about that). These two areas are the 'knowns' and, to get insights about the product, this can be handled by running (automated) checks.

There are also things that we do not know we know (intuition) and things we do not know that we do not know (exploration). These two areas are the 'unknowns' and, to get insights about the product, this can be handled by exploratory testing.

By exploring, you can bring 'unknows' to the 'knowns' area.

5.3 Dimensions of Exploratory Testing

There are several dimensions **[ET1]** you can use to make Exploratory Testing more concrete. The dimensions are listed here, however its exact use is always context dependent.

5.3.1 General dimensions

LO-5.1	K2	Summarize the general dimensions of Exploratory Testing

Some dimensions that can help in exploratory testing of any application are:

- Never and Always conditions: work together with your stakeholder and make a list of things that should never and should always happen in your system.
- Identify external resources: industry standards, legislation, general reviews on comparable software.
- Identify useful approximations:
    - look at parts of the software of which you are unsure how to validate if the software is giving the correct results.
    - Find people who can tell you the correct results.

5.3.2 Sequences and Interactions

LO-5.2	K2	Summarize the sequences and interactions dimensions of Exploratory Testing

To discover potentially serious problems involving the use and abuse of your software, you need to vary the way you interact with it. Instead of following a sensible sequence of actions, take steps that do not follow the expected sequence, order or rules. Instead of navigating using the same mechanisms every time, vary the paths you take through your system. Instead of using reasonable data, use unreasonable data.

- Randomly select noun/verb combinations in sequences. Some examples can be:
    - Verbs: send, receive, export, delete, save.
    - Nouns: message, header, appointment. Combine them randomly and see what happens.
- Navigate randomly through your application. Use the Back button, Undo, Cancel, shortcuts and special keys.
- Use the software from different persons' perspective by using personas.

5.3.3 Entities and their relationships

LO-5.3	K2	Summarize the entities and relationships dimensions of Exploratory Testing

Your software uses things, depends on things, and manages things. It lets users create, update, and delete things. Things are the reason your software exists.

- Find the (hidden) entities, attributes and dependencies; use the PROJECT and PRODUCT mnemonics to help you.
- CRUD (Create, Read, Update, Delete) everything with zero, one and many.
- Follow the data during its lifecycle and try to break the cycle.

5.3.4 States and transitions

LO-5.4	K2	Summarize the states and transitions dimensions of Exploratory Testing

Have you ever encountered a failure that was extremely difficult to reproduce? Perhaps you have seen a catastrophic error that happens only sporadically, or maybe you stumbled on corrupted data and could not trace the root cause.

Such defects are often triggered when something happens during a brief window of vulnerability: a moment in time when all the conditions line up just right so something can go very wrong: the system transitions from one state to another.

- State: a certain condition of the system. For example: recognized, logged out, logged in, not logged in due to error.
- Transition: the event that changes the system from one state to another. For example: authenticating, authorizing.

Explore how to get from state to state and find more ways to reach a certain state. Also interrupt state transitions by logging out, killing processes, unplug power or network.

5.3.5 The ecosystem

LO-5.5	K2	Summarize the ecosystem dimensions of Exploratory Testing
HO-5.1	HO-2	Create ET test charters using the different dimensions of Exploratory Testing
HO-5.2	HO-2	Execute the created test charters to reveal new insights

Software never exists in isolation. It runs on an operating system. It depends on libraries of reusable code or external services. It uses system resources like memory, the file system, databases, and network connections. It interoperates or integrates with other applications.

- Interfaces: interfaces usually cause issues. Unclear agreements or specifications, or interpretation of specifications.

- External dependencies: unknown dependencies usually cause issues. Use PROJECT to get insights on external dependencies.
- Internal components and logs: it's not only the visible parts of the software that cause issues. Check the non-visible internal components as well, and always check all the logs for unexpected messages. Logs can reveal potential or not directly visible issues.
- What if: always ask yourself this question: what if the interface is offline? What if this file is locked? What if this file is empty?

Chapter 6 – Negotiations

Negotiation is basically a means for you to obtain what you need, as soon as you deal with a fellow human. Based on the unique position a tester has within an agile team, they are often in great need to get their perspectives understood and the needed requirements fulfilled to get the job done. Therefore, being able to execute a smooth negotiation is essential to obtain what is needed, while maintaining a healthy relationship with the team and stakeholders outside the team.

Keywords

Positional negotiation, Leary's rose, symmetric principle, complementary principle, principled negotiation, PIOC (people, interests (whishes, sorrows, needs), options, criteria (boundaries, procedures, ruling), feelings, trust, bias, BATNA, summary, questions, alternatives, accusation audit, empathy, adaptation, summarizing, asking, 'don't split the difference', 'go to the balcony'.

LO-6.1	K2	Understand the 2 principles that are leading in Leary's rose.
LO-6.2	K2	Understand the difference between "first person" and "third person" view.
LO-6.3	K2	Understand the core difference between positional vs. principled negotiation.
LO-6.4	K2	Understand the "second person"-view.
LO-6.5	K2	Understand how communicating interests (sorrows, wishes, needs) gives a clear view on the problem and creates multiple options to solve the problem.
LO-6.6	K2	Understand how applying these categories create more objectivity.
HO-6.1	HO-2	Execute a principled negotiation.
HO-6.2	HO-3	Execute a negotiation called 'the ultimatum game'.
LO-6.7	K1	Memorize that people take decisions on an emotional, not a rational basis.
LO-6.8	K1	Memorize the biases concerning people we negotiate with.
LO-6.9	K2	Understand that trust is a prerequisite to the communication of interests.
LO-6.10	K2	Understand the limits of the use of objective criteria.
LO-6.11	K2	Explain the power of preparing questions and an accusation audit.
LO-6.12	K2	Explain how these elements create a relationship with an interlocutor.
LO-6.12	K2	Explain what the effect is of these four elements.
LO-6.14	K2	Understand how these elements influence the negotation process.
HO-6.3	HO-1	Execute a negotiation with humans.

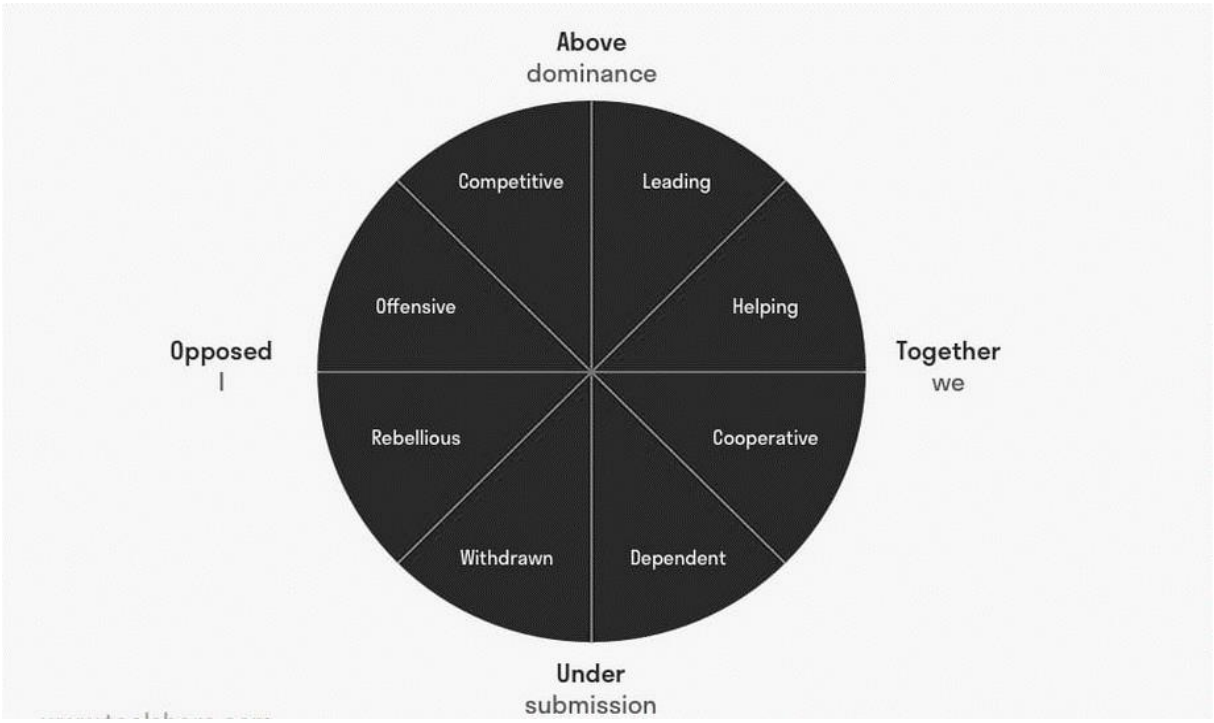
6.1 The definition of negotiation and our positional behavior

LO-6.1	K2	Understand the 2 principles that are leading in Leary's rose.
LO-6.2	K2	Understand the difference between "first person" and "third person" view.

Negotiating is basically 'a game' where collecting information and influencing behavior takes place. Negotiation is "result driven communication": we want something, and we are depending on our fellow humans to obtain it. Based on those facts, we take position relative to our interlocutor.
A common problem is that we all too often take only our own point of view ("first person view") into account, choosing avoidance, adaptation, competition or compromising. This way, we seldom come to the most promising outcome of a negotiation: the win-win situation.

Another way to look at negotiation is to use the rose of Leary **[NE1]**.

Figure 6.1



When you look at a negotiation from a distance (a 'third person view') you notice that the two parties involved in a negotiation are moving like in a dance: when one party shifts position, so does the other one. For example: when one of the interlocutors becomes "offensive", chances rise that the other party shifts position from, let's say "cooperative", to "rebellious" or "withdrawn". When a new shift takes place, let's say from "offensive" to "dependent", the other party shifts to "helping" or "leading". These shifts take place due to two principles: the symmetric (if you oppose me, I will oppose you / if you help me, I will help you) and the complementary principle (If you go above, I go under / if you go under, I go above).

6.2 Principled negotiation

LO-6.3	K2	Understand the core difference between positional vs. principled negotiation.
LO-6.4	K2	Understand the "second person"-view.
LO-6.5	K2	Understand how communicating interests (sorrows, wishes, needs) gives a clear view on the problem and creates multiple options to solve the problem.
LO-6.6	K2	Understand how applying these categories create a more objective playground.
HO-6.1	HO-2	Execute a principled negotiation.

Positional negotiation (the "first person view") very often becomes emotionally driven, which can thus easily lead to a conflict situation in which the people involved become trapped.

To prevent this from happening, Harvard developed the principled negotiation method **[NE2]**: rationality must be the leading element in a negotiation, whereby the negotiation is positioned as a

problem (versus a fight) and the parties involved are problem solvers (versus adversaries). You can look at this as a combination between the “first person view” and the “second person view”: although taking own interests into account, also understanding and dealing with the interests of your interlocutor is vital to come to a “win-win”-situation. To do this, 4 rules apply: one should divide the person from the problem, look for interests (sorrows, wishes, needs) instead of standpoints, look for options to solve the shared problem and use objective criteria (boundaries (law, money, market), procedures (letting fate decide, or experts) and ruling (agree that logical reasoning must be used and come to an agreement concerning process time)) when the situation becomes more complicated.

6.3 The limitations of a principled negotiation

HO-6.2	HO-3	Execute a negotiation called ‘the ultimatum game’.
LO-6.7	K1	Memorize that people take decisions on an emotional, not a rational basis.
LO-6.8	K1	Memorize the biases concerning people we negotiate with.
LO-6.9	K2	Understand that trust is a prerequisite to the communication of interests.
LO-6.10	K2	Understand the limits of the use of objective criteria.

Although the principled negotiation method has great advantages over a more positional negotiation, the problem is that people are basically not rational, but rather emotional when a decision is made **[NE3]**. The ‘ultimatum game’ demonstrates this **[NE4]**. Also, the building blocks of principled negotiation have their flaws. Are we able to separate the problem from the person in front of us? The halo and horn biases make that almost impossible. Can we focus on interests, if there is too little trust? And what about objective criteria or logical reasoning? Humans make decisions rather based on feelings like connection, freedom, trust or on what we consider fair, then on rational criteria.

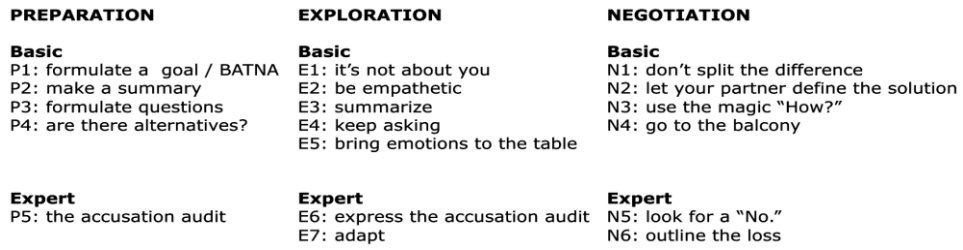
6.4 Negotiating with humans

LO-6.11	K2	Explain the power of preparing questions and an accusation audit.
LO-6.12	K2	Explain how these elements create a relationship with an interlocutor.
LO-6.14	K2	Explain what the effect is of these four elements.
LO-6.14	K2	Understand how these elements influence the decision-making process.
HO-6.3	HO-1	Execute a negotiation with humans.

In the 2010s, FBI negotiator Voss **[NE5]** came up with ways on negotiating that combines the principled method with the hard-wired irrational elements of human behavior. For software testers, to be able to focus on the interests of any stakeholder in the software development process, as well as take care of their own interests, these ways are vital.

An oversight on all ways is displayed below.

Figure 6.2



Preparation

P1: formulate a goal: what do you want out of the negotiation?
P1A: formulate a BATNA: BATNA stands for "Best Alternative To a Negotiated Agreement": what can you achieve concerning the problem without negotiating at all?
P2: make a summary: create a picture on what both parties want, and the elements that dictate or can possibly change that need.
P3: formulate questions: "What are you trying to achieve? How would you like to do that?"
P4: are there alternatives?: f.e. if your price is too high, maybe the other party can pay in a service.
P5: the accusation audit: a way to create trust, is to accuse your own behavior or position. F.e.: "You might think that, based on what you saw of me until now in other situations, that I'm stubborn. You are right, I really can be. But I only behave that way when I feel misunderstood and trapped in that experience. You can help me then by asking me if I feel misunderstood, and what the reasons are."

Exploration

E1: it is not about you: focusing on the (non-)verbal communication of your interlocutor brings your own mind, needs, sorrows to rest and broadens the channel of information.
E2: be empathetic: stand in the shoes of your interlocutor, take the "second person"-view.
E3: summarize: if you summarize what your partner has said, he/she will feel more understood.
E4: keep asking: to get a clear picture, one should push for the root of the problem.
E5: bring emotions to the table: negative emotions tend to undermine the negotiations if they are not put into the open. If you notice resentment, anger, fear, mention them and let your interlocutor elaborate.
E6: Express your accusation audit.: express what you have prepared (P5).
E7: adapt: Humans trust themselves. If you 'talk and walk' like your interlocutor, trust rises.

Negotiation

N1: don't split the difference: in a negotiation, humans tend to go for a compromise to get rid of the nasty tension. Suppress that feeling; you will not realize your goals, or those of your interlocutor.
N2: let your partner define the solution: humans need control. If they come up with the solution, they are more empowered to back up that solution then when someone else defines it.
N3: use the magic "how"?: when your partner pushes for a solution that lays a great burden on you ("you must test these 500 cases before the week is over!"), asking "How?" involves her or him.
N4: "go to the balcony.": if things heat up, and fierce emotions interfere, interrupt the negotiation. While being alone, remember yourself why a succesful outcome of the negotiation is important.
N5: look for a 'No': when humans say 'no' they draw a line, what makes them feel secure.
N6: outline the loss: paint the picture of the loss for you and your partner if the negotiation is unsuccessful.

Chapter 7 – Visualization

Most people tend to create large chunks of text to make a specific point. In a scrum team, people like to create user stories, a test strategy, test cases, risk lists, improvement lists, ….

It is also known that most people are not the best at reading all this text. Opting for more visual information instead, will be more effective.

Keywords

Visualization, insights, break down, cooperation, visual thinking

LO-7.1	K2	Demonstrate the value of visualization
HO-7.1	K3	Practice your basic visualization skills
LO-7.2	K2	Explain why visualization should be used
LO-7.3	K2	Demonstrate when visualization can be used
LO-7.4	K1	Recall the visualizations on project & product outline and risks
LO-7.5	K2	Demonstrate PopcornFlow© as a visualization tool
LO-7.6	K2	Demonstrate subway maps as a visualization tool
HO-7.2	K3	Practice visualization with one of the tools for a given problem

7.1 Why do people need visualization

LO-7.1	K2	Demonstrate the value of visualization
HO-7.1	K3	Practice your basic visualization skills
LO-7.2	K2	Explain why visualization should be used

In order to demonstrate the value of visualization, a little exercise is started in which participants visualize how to make toast (the bread). After the participants have demonstrated their result a video of Tom Wujec **[VZ1]** is shown that explains what has just happened.

Visualization of in this case toast comes up with many different drawings in which different angles play a role: the toast, the toaster, the mechanics of the toaster, the people experience or the whole supply chain of toast. They have 2 elements in common:

- Nodes (the pictures of the toast, toaster, people, …)
- Links between the nodes

All pictures are system models and represent a private mental model with nodes and links.

If the models are discussed in the group, you can create a new model with the whole group so in the end everybody has the same point of view on, in this example, making toast.

Visualization can help you because of a number of reasons:

- You can get a quick insight into a specific problem
- You can break down a big problem into smaller pieces
- It makes it easier to work together as a team on a specific solution
- Many people are visual thinkers

- One picture can say more than a thousand words

7.2 When do people need visualization

LO-7.2	K2	Explain why visualization should be used
LO-7.3	K2	Demonstrate when visualization can be used

Visualization can be useful in a number of situations:

- To get overview in architecture of the environment or application
- During retrospectives on improvements on the process and team collaboration
- During testing when you run into an issue on the issue itself and how you got there
- When creating your test strategy by means of the project outline, product outline and potential damages
- When explaining your test strategy to the team or stakeholders

7.3 How do people visualize

LO-7.4	K1	Recall the visualizations on project & product outline and risks
LO-7.5	K2	Demonstrate PopcornFlow© as a visualization tool
LO-7.6	K2	Demonstrate subway maps as a visualization tool
HO-7.2	K3	Practice visualization with one of the tools for a given problem

The mind maps (or other forms of visualization) are recalled that were created in the AU-CPAT training on the project outline, the product outline and the risks.

Besides mind mapping that is explained in the AU-CPAT, two other visualization tools are demonstrated that are specifically useful to visualize improvements.

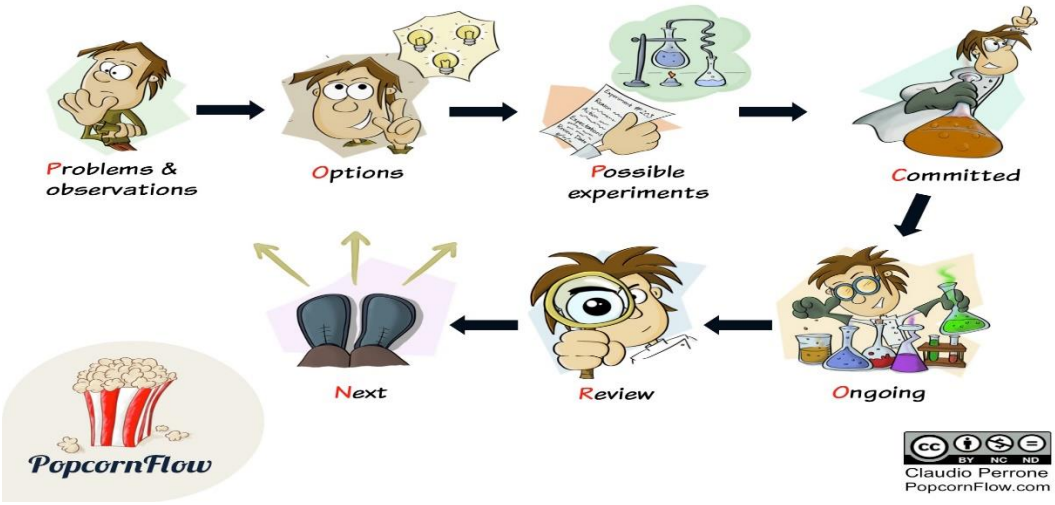
7.3.1 PopcornFlow©

PopcornFlow© **[VZ2]** is a visualization method/tool to help your team move fast, learn faster and thrive via ultra-rapid experimentation.

PopcornFlow© consists of 7 stages:

- Problems & observations
  - What is the problem, what challenges do you have?
- Options
  - What options could there be to solve a specific problem?
- Possible experiments
  - What experiments could be performed based on the options?
- Committed

    - Which experiments are being performed and what expectations are there?
- Ongoing
    - Which experiments are currently in progress?
- Review
    - How did the experiment go? Did the experiment meet expectations, or did it fail and why?
- Next
    - Is the problem solved and can the team continue with the next one?
    - Or should the team pick another experiment in trying to solve the issue?
    - Or should the team consider another option to solve the problem?

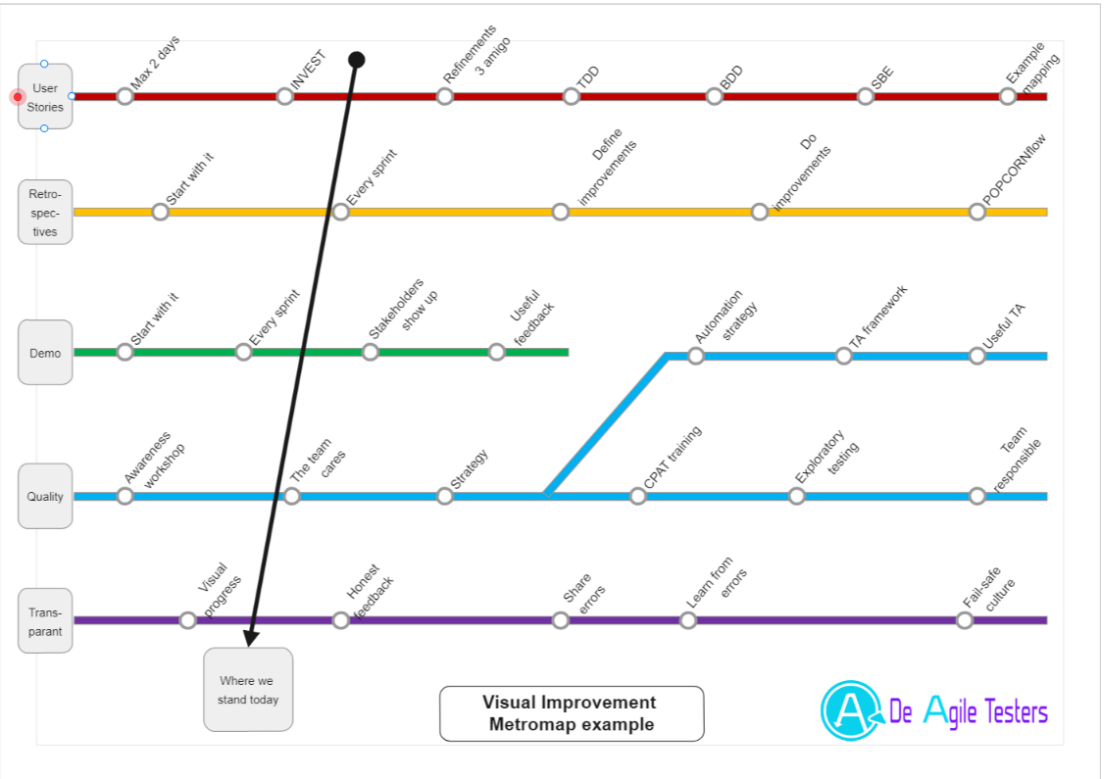


An example:

1. Problem: our code quality sucks
2. Options: TDD/BDD, SonarQube, Pair Programming
3. Possible Experiments: 3 day Pair programming, Read BDD Book
4. Committed: Pair programming. Expectations: we write better code, less bugs, we like it
5. Ongoing: f.e. our 'storytelling' experiment day 3 of 5 and code quality day 1 of 3
6. Review (in this example after 3 days of pair programming: 3 developers loved it, 1 hates it, we create 70% less bugs, we get 80% less SonarQube errors. We want to continue.
7. Next: next problem or next option or next experiment (depending on the result of the previous one).

7.3.2 Subway maps

Another way to visualize improvements is to use a subway map **[VZ3]**. The metro route is the way to improvement (from starting station to the end station) and each station on the way is a next step in the improvement. The line marks where you stand today and will gradually move to the right if you actually improve. Each step in the improvement could in more detail be done using PopcornFlow©.



References

Reference	Source
[BT1]	A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives - By L. Anderson, P. W. Airasian, and D. R. Krathwohl (Allyn & Bacon 2001)
[BT2]	https://www.apu.edu/live_data/files/333/blooms_taxonomy_action_verbs.pdf
[IN1]	Agile Testing, A practical guide for testers and agile teams https://agiletester.ca/ More Agile Testing, Learning Journeys for the Whole Team https://agiletester.ca/
[LS1]	Deci, E. L., & Ryan, R. M. (2012). Motivation, personality, and development within embedded social contexts: An overview of self-determination theory. In R. M. Ryan (Ed.), Oxford handbook of human motivation (pp. 85-107). Oxford, UK: Oxford University Press
[LS2]	https://www.ted.com/talks/amy_edmondson_how_to_turn_a_group_of_strangers_into_a_team
[LS3]	"Psychological safety Trust, and Learning in Organisations. A group level lens by Amy C. Edmondson
[LS4]	"How to build and rebuild trust" Francis Frei, TED
[LS5]	C.G. Goulding 6 Practical Ways to Create an Accountability Culture in a Company (lifehack.org)
[LS6]	Bev Atfield: 7 ways to create psychological safety at work (jostle.me)
[LS7]	https://cassierobinson.medium.com/a-user-manual-for-me-d3a851fbc694
[PD1]	https://www.cgerisk.com/knowledgebase/The_bowtie_method
[PD2]	A new perspective on how to understand, assess and manage risk and the unforeseen - ScienceDirect - Aven and Krohn
[BH1]	Presentation ´Bug hunt´ by Klaus Olsen, Test Adviser, Softwaretest.dk ´Bug Hunt: Making Early Software Testing Lessons Engaging and Affordable´, Sebastian Elbaum, Suzette Person, Jon Dokulil, Computer Science and Engineering Department, University of Nebraska-Lincoln, Lincoln, Nebraska, USA
[BH2]	Soap Opera scenario's: Hans Buwalda(published in the February 2004 issue of *Better Software*)
[BH3]	https://en.wikipedia.org/wiki/Crowdsourced_testing
[BI1]	William Hazet (10/04/1778 – 18/09/1830). English writer.
[BI2]	https://www.merriam-webster.com/dictionary/bias
[BI3]	Thinking, Fast and Slow, D. Kahneman, ISBN: 9780141033570
[BI4]	https://www.pon.harvard.edu/daily/negotiation-skills-daily/the-drawbacks-of-goals/
[BI5]	https://en.wikipedia.org/wiki/Framing_(social_sciences)
[BI6]	https://en.wikipedia.org/wiki/Bias#Halo_effect_and_horn_effect https://study.com/academy/lesson/the-halo-effect-definition-advantages-disadvantages.html
[BI7]	Statistics workbook for dummies, Deborah J. Rumsey PhD. ISBN 978-1119293521 https://www.dummies.com/education/math/statistics/how-to-identify-statistical-bias/
[BI8]	Klaus Conrad: Die beginnende Schizophrenie. Versuch einer Gestaltanalyse des Wahns (1958) https://en.wikipedia.org/wiki/Apophenia http://skepdic.com/apophenia.html

[BI9]	Lo, Bernard; Field, Marilyn J. (2009). Conflict of Interest in Medical Research, Education, and Practice. Washington, D.C.: National Academies Press. ISBN 978-0-309-13188-9. https://en.wikipedia.org/wiki/Bias#Conflicts_of_interest
[BI10]	Heider, F. (1958). "The psychology of interpersonal relations", New York: Wiley https://en.wikipedia.org/wiki/Attribution_bias
[BI11]	Nickerson, Raymond S. (1998), "Confirmation bias: A ubiquitous phenomenon in many guises", Review of General Psychology, 2 (2): 175–220, doi:10.1037/1089-2680.2.2.175, S2CID 8508954 https://en.wikipedia.org/wiki/Confirmation_bias
[BI12]	Kahneman, D.; Knetsch, J. L.; Thaler, R. H. (1991). "Anomalies: The Endowment Effect, Loss Aversion, and Status Quo Bias". Journal of Economic Perspectives. 5 (1): 193–206. doi:10.1257/jep.5.1.193. https://en.wikipedia.org/wiki/Status_quo_bias
[BI13]	Bethlehem, Douglas W. (2015-06-19). A Social Psychology of Prejudice. Psychology Press. ISBN 978-1-317-54855-3. https://en.wikipedia.org/wiki/Prejudice
[BI14]	https://medium.com/swlh/how-to-overcome-cognitive-biases-and-make-better-decisions-daeecd38f910
[BI15]	Dunning-Kruger effect | Definition, Examples, & Facts | Britannica
[ET1]	Explore IT!, Reduce Risk and Increase Confidence with Exploratory Testing. E. Hendrickson. Version 2.0 September 2013
[NE1]	https://en.wikipedia.org/wiki/Interpersonal_circumplex
[NE2]	https://www.academia.edu/40096618/Principled_Negotiation_The_Harvard_Approach_Fisher_and_Ury
[NE3]	https://en.wikipedia.org/wiki/Descartes%27_Error
[NE4]	https://en.wikipedia.org/wiki/Ultimatum_game
[NE5]	Chriss Voss, Thal Raz: “Never split the difference. Negotiating as if your life depended on it.“ ISBN 9780062407801
[VZ1]	https://www.ted.com/talks/tom_wujec_got_a_wicked_problem_first_tell_me_how_you_make_toast
[VZ2]	https://agilesensei.com/popcornflow/ https://www.youtube.com/watch?v=cqtxMy58kz8 https://www.slideshare.net/cperrone/popcornflow-continuous-evolution-through-ultrarapid-experimentation
[VZ3]	Read more about visualization and subway maps in: Beautiful Visualization by Julie Steele and Noah Iliinsky ISBN 9781449379865