IAPM

AGILE PROJECT MANAGEMENT GUIDE 2.0

SCRUM / KANBAN
EXTREME PROGRAMMING

iapm
INTERNATIONAL ASSOCIATION OF PROJECT MANAGERS

IAPM
INTERNATIONAL ASSOCIATION OF PROJECT MANAGERS

In 1997 the IAPM was still a fledgling association. It started out as a loosely structured international network for project managers who shared the objectives of promoting and modernising project management and providing young project managers with the tools to work effectively and successfully. Since this time, the IAPM has held annual International Project Manager Meetings (IPMM). Back in 1998 the IAPM published the precursor to the PM Guide 2.0, the IAPM By-laws of Project Management. These by-laws were completely revised and adapted to modern requirements and real-life project management scenarios in the PM Guide 2.0, which was published in 2010. In the same year, the IAPM was completely relaunched. The Scrum Guide 1.0, this Agile PM Guide 2.0's precursor, was published in March 2011.

In 2012 the IAPM introduced two awards, the Project Manager of the Year Award and the Book of the Year Award.

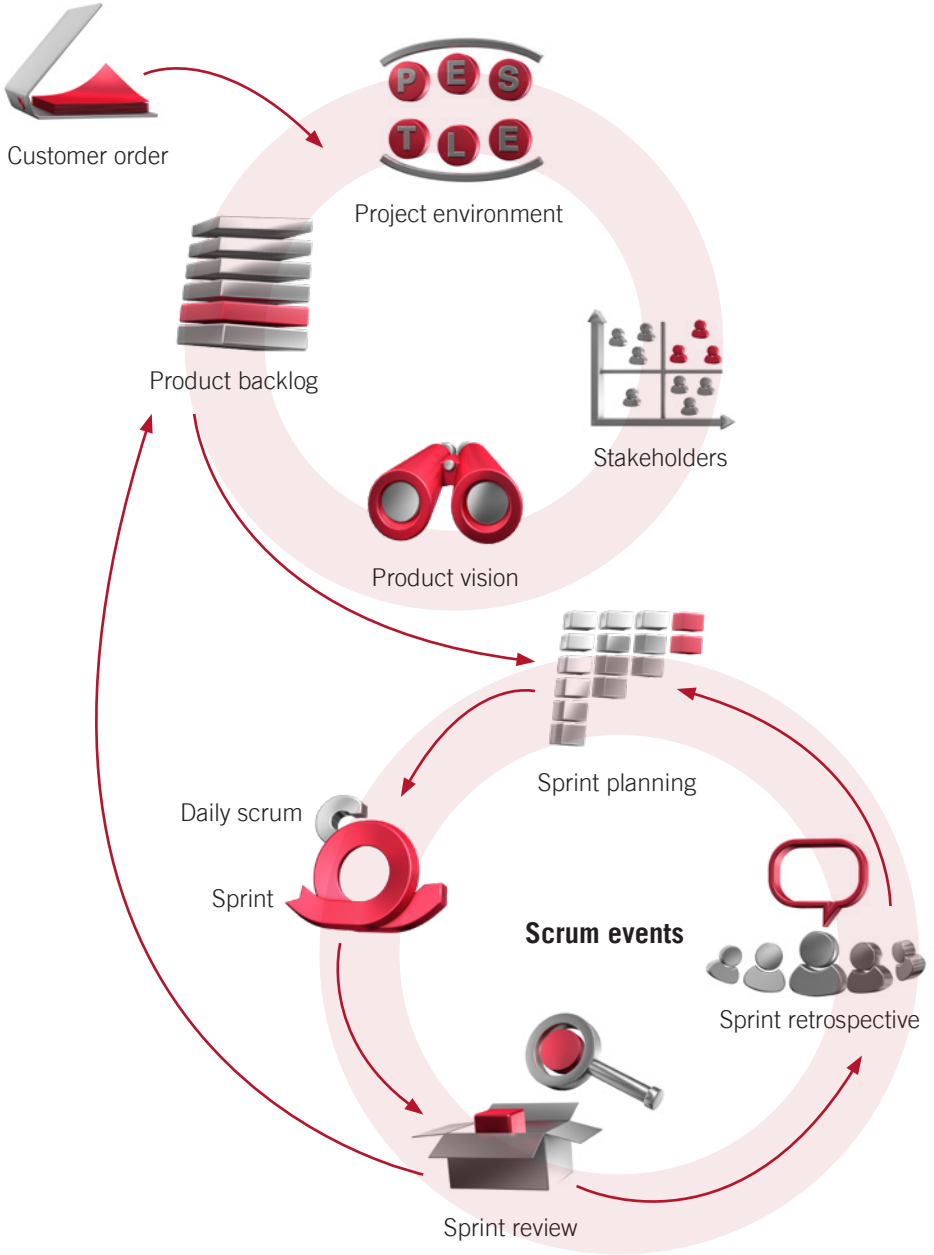
The Project Manager of the Year Award is very special to the IAPM because it pays tribute to IAPM Senior Project Managers for their outstanding achievements in project management.

The Book of the Year award honours books on the subject of project management that are published in both German and English. These books may communicate experience and knowledge in an innovative way, be (auto)biographical works or textbooks providing an introduction to the subject of project management.

The IAPM is an independent certification body which examines the knowledge and competence of the certification candidates with a comprehensive, fair and neutral online examination system. The certification system is therefore tailored to the challenging world of project management in the 21st century.

© IAPM International Association of Project Managers

PROJEKTATLAS AGILES PM



CONTENT

INTRODUCTION

WHAT IS AGILE PROJECT MANAGEMENT?

Agile project management is a generic term for various approaches that are founded in empirical process control. Agile methods are often used when the process to achieving a specific project deliverable, such as a software program, is difficult or impossible to plan at the outset of the project. They also make it possible for the customer to change or adapt the requirements of the final product during the project lifecycle.
There are many agile methods, such as Scrum, Kanban, Extreme Programming, MVP, Feature Driven Development, Test Driven Development and Crystal Clear, though Scrum is the method in the most widespread use around the world. It provides a framework offering the project team maximum flexibility in developing an optimum product within a defined timeframe and budget.

Scrum dispenses with the very comprehensive and detailed planning documents that are used in traditional project management. Instead, at the beginning of the project, the requirements that the product has to meet are **set out in a one-dimensional, prioritised list called the product backlog** for incremental implementation in **brief development iterations** called **sprints**. Sprints are a core element of Scrum which are used to create a 'done' product increment. The product increment is sent to the customer or the customer's authorised representative at the end of the sprint for qualified feedback. This feedback makes it possible to correct expensive mistakes at an early stage of the project, thereby preventing wasted time and escalating costs. Agile methods are perfect for projects with vague technical implementation objectives and incompletely formulated customer specifications.
That's why it's often impossible to predict with any certainty how long it will take to develop the finished product or what exactly the cost will be at the outset of an agile project. Although this is a concern to some organisations when they first roll out Scrum, it is still possible to estimate in a Scrum project and this guide sets out suitable methods and approaches for doing that.

It also explains the Kanban agile project management method and the Extreme Programming software development methodology.

THE FOUR PARTS OF THE AGILE PM GUIDE 2.0

This Guide will help you to make targeted use of your agile project management knowledge.

PART 1:
AGILE PROJECT MANAGEMENT - SCRUM

What are the rules in the Scrum framework?
What are roles and artefacts in Scrum?

PART 2:
PEOPLE IN PROJECTS - SOFT FACTORS

Meetings, teams, motivation and conflicts –
people as key factors in an agile environment.

PART 3:
OTHER AGILE METHODS

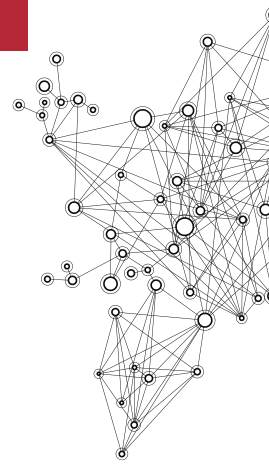
Scrum or Kanban? Or maybe Extreme Programming?

PART 4:
THE IAPM-CERTIFIED AGILE PROJECT MANAGER

How to obtain certification of your agile project management competence and enhance your market value.

Good luck with your projects
and be a great team player!

iapm
INTERNATIONAL ASSOCIATION OF PROJECT MANAGERS



PART 1 AGILE PROJECT MANAGE-MENT - SCRUM

SCRUM ROLES

THE THREE MAIN ROLES

Scrum differs from traditional project management in that there is no project manager and the team members have one of three roles.

Those three roles are Product Owner, Development Team or Scrum Master. Collectively, they are the **Scrum team**. The **Product Owner** is the customer or the customer's authorised representative in the project. He conveys the customer vision to the Scrum team and is responsible for ROI.

The cross-functional and self-organising **Development Team** works autonomously and makes its own decisions in the product development process. The **Scrum Master** is team coach and facilitator, with responsibility for optimising the team's work environment, eliminating organisational impediments and shielding the team from interruptions during a sprint.

1 PRODUCT OWNER

- Responsible for maximising the product's return on investment (ROI)
- Develops the product vision
- Represents the customer and users
- Responsible for expressing the product backlog items
- Responsible for stakeholder management
- Is ideally available to the Development Team during the sprint to answer any questions
- Responsible for accepting product increments (sprint results)
- Authority to decide whether to continue or terminate the project

2 DEVELOPMENT TEAM

- Self-organising
- Collectively responsible for development increments
- Cross-functional (with all of the skills as a team necessary to complete a product increment)
- Negotiates with the Product Owner on the scope of sprints
- Responsible for deciding how to perform the tasks in a sprint
- Should ideally be composed of 7 +/-2 members

3 SCRUM MASTER

- Responsible for ensuring that the Scrum Team understands and enacts Scrum
- Supports the Development Team's self-organisation
- Ensures that the Scrum rules are observed by the Scrum Team
- Resolves impediments and shields the team from interruptions during a sprint
- Is a facilitator, which involves preserving the integrity and spirit of the Scrum framework and making improvements when necessary
- Monitors the Development Team's performance
- Liaises with the project's organisational stakeholders
- Has no authority over the Development Team

ROLE ASSIGNMENT IN LARGE-SCALE SCRUM PROJECTS - SCRUM OF SCRUMS

In large-scale Scrum projects, it is possible and sometimes necessary to have several Development Teams working concurrently. In this case, the Scrum Master can be servant leader to several Development Teams at once. This isn't the case with the Product Owner role, however, because a dedicated Product Owner is assigned to each Development Team. In this constellation, it is important to assign a Chief Product Owner who has overall responsibility and final decision making authority.

RESPONSIBILITIES IN SCRUM PROJECTS

Item	Product Owner	Development Team	Scrum Master
Product Vision			
Product Backlog			support
Sprint Backlog			support
Sprint Burndown Chart			support
Release Burndown Chart			support
Product Increment			
Release Planning			support

ATTENDANCE OF SCRUM MEETINGS

Meetings	Product Owner	Development Team	Scrum Master
Sprint Planning Meeting			
Daily Scrum			should attend
Sprint Review			
Sprint Retrospective			
Backlog Grooming*			should attend

**Backlog Grooming isn't an official Scrum meeting, but it can help to increase team productivity*

PRODUCT VISION

THE DEVELOPMENT COMPASS

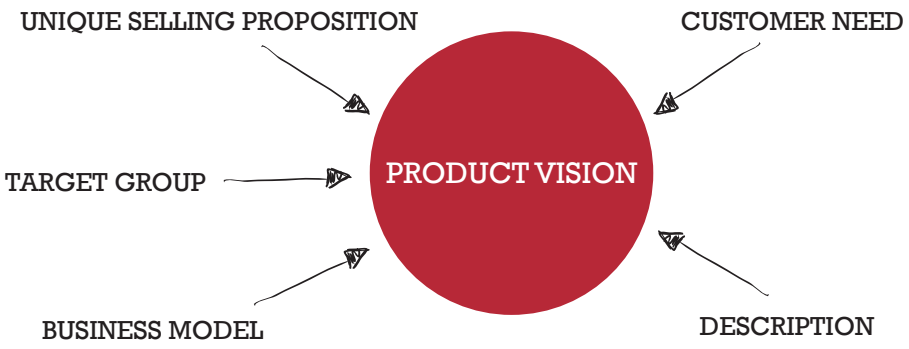
The product vision is the compass for the development of the new product. The timeframe or horizon extends far beyond the scrum project's close-out date and encompasses the entire product lifecycle. The objective is to formulate the desired commercial success as a vision and as objectives, and to consider how it can be achieved.

This is the responsibility of the product owner.

It's always a good idea to involve customers and users in the vision development process. The product owner achieves consensus between the project stakeholders on the project objective and establishes the framework for marketing the product and for its commercial success.

The following questions have to be answered:

- Which target group has to be addressed?
- What need does the customer/ user have, or what benefit is being provided?
- What is the product? What exactly does it offer?
- What USPs does the application or product have compared with competitor products?
- How is the business model structured? How will it generate income?



Analysing and managing the environment and the stakeholders are important aspects of planning and implementing any agile project.

PROJECT ENVIRONMENT

Projects aren't implemented in a vacuum. They are implemented within legal/contractual frameworks and subject to constraints on human/technical resources etc. The timely identification of important issues so that they can be taken into account enables the project manager ensure that the project is a success. What kind of an environment does the project have? What things have to be paid attention to in the following areas?

P

Politics - political framework
Which "powerhouses" have to be taken into consideration? Are there conflicts of interests?

E

Economics - economic framework
Are there economic constraints? Important economic interests? Competitors? Do seasonal or cyclical fluctuations have to be taken into account?

S

Sociology - sociological framework
Is the project subject to ethical or moral constraints? Do you have to take the sentiments or emotions of people affected by the project into account?

T

Technology - technological framework
Do technical innovations have to be integrated in the project? Are the technologies tried and tested? Do trade mark rights or licenses have to be taken into account?

E

Environment - ecological framework
Does the project pollute the environment? Do environmental regulations or restrictions have to be taken into consideration?

L

Law - legal framework
What is the legal framework? Which laws and regulations apply to the project?

STAKEHOLDERS

Stakeholders are affected by or have influence over a project. They can be either positively or negatively disposed towards it. That's why it is important to analyse the project stakeholders. The better you know them, the better you can anticipate their reactions. The stakeholder strategy addresses project stakeholders and gets them involved in designing the planned product. Both scrum masters and product owners should consider their stakeholders. Who are the stakeholders from a product owner's and a scrum master's perspective?

From the product owner's perspective:
customer, project sponsor, users, developers, sales and marketing personnel, suppliers etc.

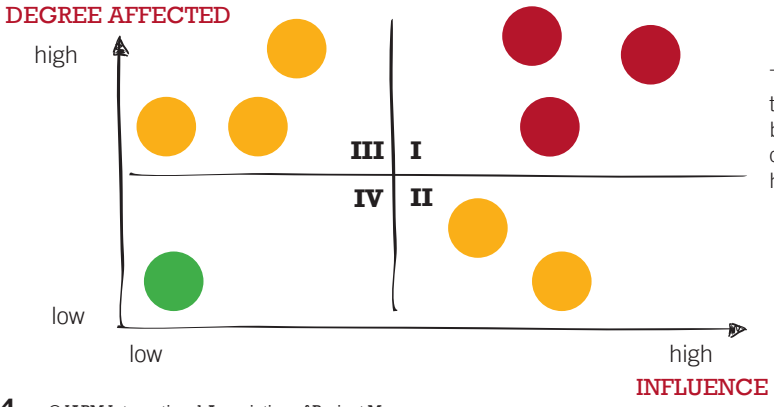
From the scrum master's perspective:
All persons or departments/units that are involved in the project, in the business environment, company executives, development team managers, line organisation, HR department, procurement team, works council etc.

The following steps have to be performed:

1. Analyse the stakeholders or stakeholder groups to determine their influence and how they are affected by the project.

2. Develop a strategy on how to deal with the stakeholders in the project (provision of information on a portal or in a newsletter, build a core team, face-to-face contact, invitations to sprint reviews).

Stakeholder analysis chart



The chart indicates how the stakeholders should be involved in the project organisation and stakeholder communication.

STAKEHOLDER AND PROJECT ENVIRONMENT MANAGEMENT

There are various options available to the product owner and scrum master for project environment and stakeholder management. They include, for example (depending on project size):

Product owner options
Attendance of sprint planning meetings and sprint reviews, establishment of an external workgroup, continuous information via newsletter, creation of a project portal.

Scrum master options
Networking in the organisation, personal contact with key individuals in the organisation, establishment of a workgroup to roll out Scrum in the organisation, creation of an information portal, consultation with executive managers about necessary actions.

PRODUCT BACKLOG

REQUIREMENTS LIST

The product backlog is created at the outset of the project. It is an ordered list of everything that might be needed in the product and the Product Owner is responsible for it.

The product backlog is a one-dimensional, prioritised list of user stories.
User stories are short descriptions of product features from the user's perspective that are written by the Product Owner. They don't make any reference to how the features are to be technically implemented - this comes later in the sprint backlog.

As soon as the first product backlog has been created, the user stories are organised in order of priority. The user story at the top of the product backlog is the one that the Product Owner wants implemented first. The main criterion for prioritisation is generally the business value generated for the user.

During a sprint, the Development Team commits to completing a specific number of user stories from the top of the product backlog to produce a product increment. This is only possible if the user stories at the top of the product backlog are designed for implementation in brief sprint iterations.

Further down the product backlog are more stories of different sizes. There will be other user stories, epics (large user stories) and themes (a collection of related user stories).

As soon as epics and themes move up to the top of the product backlog, they have to be split into multiple smaller stories so that the team can work on them.

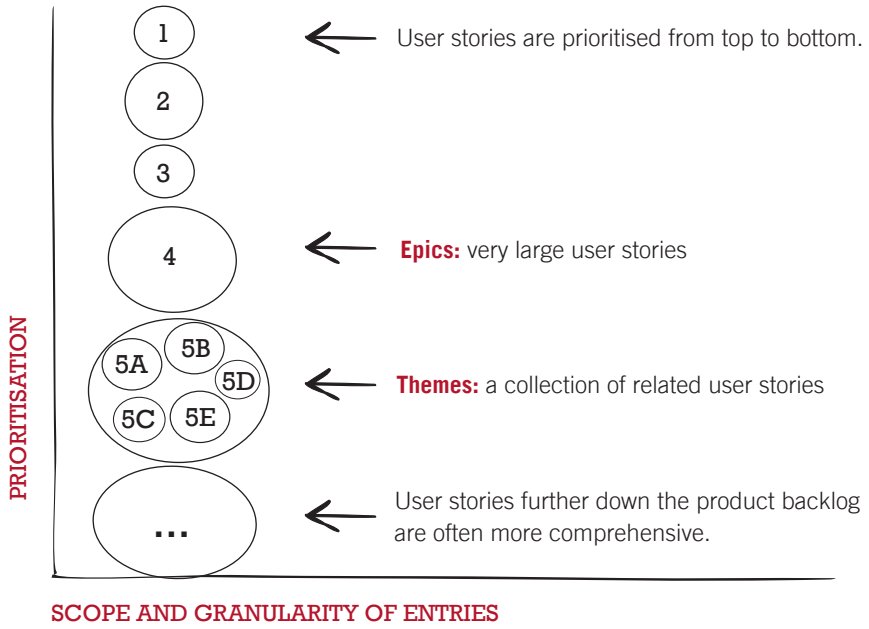
When the Development Team commits to a user story it is taken out of the product backlog and the other stories move up a place.

So the product backlog is dynamic. Product backlogs can also change due to other developments such as new user stories being created on the basis of previous increments, or priority shifts as a result of changing parameters, or the removal of user stories that have been identified as irrelevant from the product backlog.

The product backlog

- Is the Product Owner's responsibility.
- Contains a list of features and functions required to complete the product.
- Mainly consists of user stories.
- Contains user stories that are estimated and prioritised in terms of importance (business value contribution).
- User stories with the highest priority are dealt with in the next sprint.
- The product backlog is regularly updated by the Product Owner during the project.
- It is a dynamic and central aspect of agile adaptation.
- The product backlog must be DEEP (Detailed Appropriately, Estimated, Emergent, Prioritised).

PRODUCT BACKLOG ILLUSTRATION



USER STORIES

ITEMS IN THE PRODUCT BACKLOG

The customer or user's perspective of the product is an important criterion in the creation and updating of the product backlog. User stories give users the opportunity to communicate what they want in their own words.
That's why the backlog items are always created in user story format. In the early days of Scrum, technical implementation details were also included in the product backlog, though this is not the norm today. Technical aspects and implementation issues are included in the sprint backlog (see page 21).

A user story has three parts: a 'role', a 'goal or desire' and a 'benefit' (i.e. the business value delivered from the user's viewpoint). This ensures that all project participants - customer, user and other stakeholders - can express what they would like to see in the product without technical expertise or being able to express themselves in technical terms.

To ensure that a user story is implementable, it should satisfy the **INVEST** criteria:

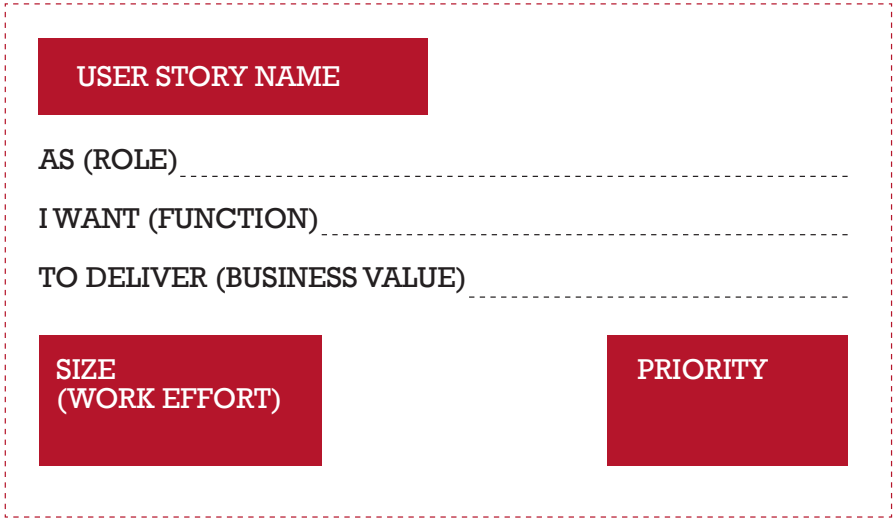
Independent
Negotiable
Valuable
Estimable
Short
Testable

It is also important to remember the **three Cs –** Card, Conversation and Confirmation – when including user stories in the product backlog.

Card – User stories are written on cards. The card contains notes on priority, work effort, acceptance criteria and estimated work effort of implementation.

Conversation – The Product Owner and the Development Team discuss the user story in detail so that everyone understands the goal.

Confirmation – Acceptance criteria are defined for the Product Owner's acceptance tests confirming implementation of the user story in the product increment.



GROOMING THE PRODUCT BACKLOG

The product backlog should be reviewed and evaluated, i.e. 'groomed' at regular intervals. Grooming involves going through the stories in the backlog to check whether they are still relevant and whether they still reflect stakeholder interests. This takes place at the **backlog grooming meeting.**

It's always important to have an adequate number of user stories with a **Definition of Ready**, i.e. user stories that are immediately actionable. Product backlog items that are ready meet the three Cs and the INVEST criteria, and are small and detailed enough to be implemented in tasks. So 'ready' stories can be included in the sprint planning process and assigned to sprints.

AGILE ESTIMATION

ESTIMATING WORK EFFORT

Estimating the work effort necessary to implement user stories in the product backlog is a crucial aspect of agile project planning. It gives the Scrum team an idea of the total scope of work over the course of the project.

No specific estimates, such as work effort in terms of working hours, are made. The objective of backlog estimating is to look at the backlog items in relation to one another. Product backlog estimates make it easier to adapt release plans to changes that affect velocity (see page 24), and to assess the impacts of these changes.

Agile estimation

- The team uses a method such as 'planning poker' to do the estimate.
- Every item in the product backlog is estimated.
- The team starts off with the item that is expected to require the least work.
- Then all other items are compared with the first item.
- Story points are awarded to rate the relative work effort.
- It's a good idea to award points in a Fibonacci-like format. The first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two (1,2,3,5,8,13,21,34). Items at the top end of the sequence are epics and will have to be split into multiple user stories.
- Another method is to compare user stories to dress sizes, i.e. XXS, XS, S, M, L, XL and XXL. Items at the bottom end of the scale are epics and will have to be split into multiple user stories.

SPRINT BACKLOG

WHAT ARE WE DOING NEXT?

At the beginning of each sprint, the team selects the items in the product backlog that have to be processed into a deliverable (= tested & presentable) product in that sprint. This takes place at the sprint planning meeting, which is attended by the Product Owner, the Scrum Master and the Development Team. The Development Team decides how many items will be processed in the next sprint based on the velocity (the number of user stories completed) of the previous sprint.

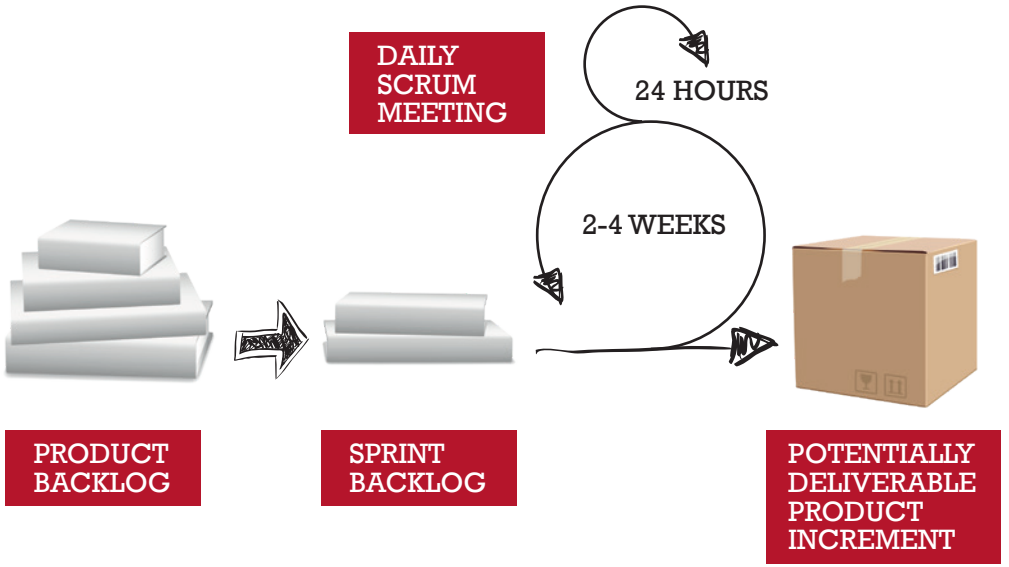
It has proven useful to have a task board positioned in the development area where everyone can see it showing the sprint backlog. The user stories are listed in order of priority in the left-hand column. The column to the right of the user stories contains a to-do list of tasks associated with each user story in the product increment. These tasks are defined by the Development Team in the sprint planning meeting. The next column lists work in progress, and the last one lists tasks that have been done. This ensures that everyone can see the current status of all the tasks. If the Development Team is distributed across different locations, it's a good idea to use a software-based sprint backlog.

The sprint backlog...

- is the list of items to be completed in the current sprint.
- describes the tasks to be implemented (e.g. design, architecture, programming, testing and refactoring).
- is the responsibility of the Product Owner and the Development Team, and prepared at the sprint planning meeting.
- is displayed in the development area where everyone can see it. The sprint backlog is updated before each daily scrum meeting (daily stand-up meeting).

SPRINTING

Sprints are the basic units of development in an agile project. During each sprint, the team creates finished portions of the product. A sprint is a 'timeboxed' effort, i.e. it is restricted to a specific duration, which can vary between two and a maximum of four weeks. If possible, the sprint duration should not be changed during the project. Short sprints are a better choice for complex development projects because corrective action can be taken faster if project veers off course. The sprint backlog is updated daily during the sprint. Both the sprint backlog and the sprint burndown charts are positioned in the work area so that all members of the team can see them and keep up to date on the precise status of the sprint.

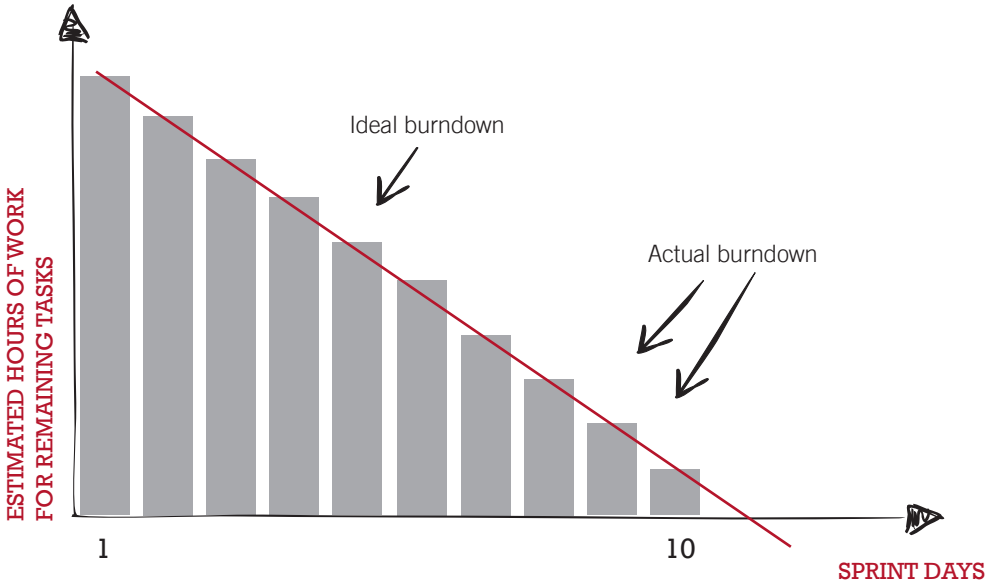


SPRINT BURNDOWN CHART

SPRINT TRACKING TOOL

The sprint burndown chart shows which tasks still have to be done and the Development Team's progress in the sprint. Each workday the estimated work effort required to perform the remaining tasks is entered in the chart and checked against the pre-plotted ideal burndown line. If the team doesn't have enough working hours remaining in the sprint to complete all the tasks, the number of tasks in the next sprint is reduced. Vice-versa, if there is time left over at the end of the sprint, the number of tasks is increased.

A sprint burnup chart is another alternative. This is an upward trending chart comparing the amount of work completed against the total amount of work.



VELOCITY

THE AMOUNT OF WORK DONE IN A SPRINT

Velocity measures the number of story points completed by a team in a sprint. Assuming that team composition and sprint length remain unchanged, there will be a new and valid velocity value after every sprint. The average velocity value is used to substantiate the release plan with the help of a release burndown chart. Velocity is always dependent on several factors, such as the project, tasks and the Development Team, so the velocity values for one project won't necessarily apply in another project.

DEFINITION OF DONE

One important thing about a sprint is that it is generally only successful if all user stories are completed in the iteration for presentation at the sprint review. There are strict requirements to be met in the sprint before the **'Definition of Done'** (DoD) applies.

'Definition of Done' is a list of activities or criteria that have to be performed or met so that the user stories in any one iteration can be defined as 'done'. The list is created by the Development Team and the Product Owner, and the content of the list always adds value to the product. Examples of DoD are:

- The acceptance criteria for the completed user stories are met.
- The release documentation is ready.
- The product increment has been tested.
- Guidelines and standards have been complied with.

Only user stories that are 100% done are included in the product increment's sprint review meeting.
Tasks that aren't done or are not done properly won't be accepted by the Product Owner in the sprint review and are **returned to the product backlog** to be redone in the next sprint. This additional development work is referred to in agile project management as **'technical debt'**.
Growth in technical debt with each new product increment increases the product's complexity and reduces velocity.

PRODUCT INCREMENT

The goal of every sprint is to deliver a tested, usable and sometimes even a marketable product increment that is accepted in the sprint review by the Product Owner and can possibly also be inspected by the stakeholders attending the meeting with the Development Team. The product increment is the deliverable when all the product backlog items in a sprint have been implemented, and the final product is the sum of all product increments.

RELEASE PLANNING

BASED ON THE PRODUCT BACKLOG

At the beginning of agile projects, planning is often only possible to a limited extent.

- The product backlog includes many items that are subjective and have to be adapted over time.
- During the project, new items will be added and existing items that have become obsolete will be deleted.
- The velocity at which the Development Team can complete user stories is not known at the outset of the project and can only be estimated.

Release planning for all estimated items in the product backlog is possible as soon as the Development Team's velocity is known. If a team has only just been formed or it is the organisation's first agile project, release planning isn't possible until the project is underway.

RELEASE BURNDOWN CHART

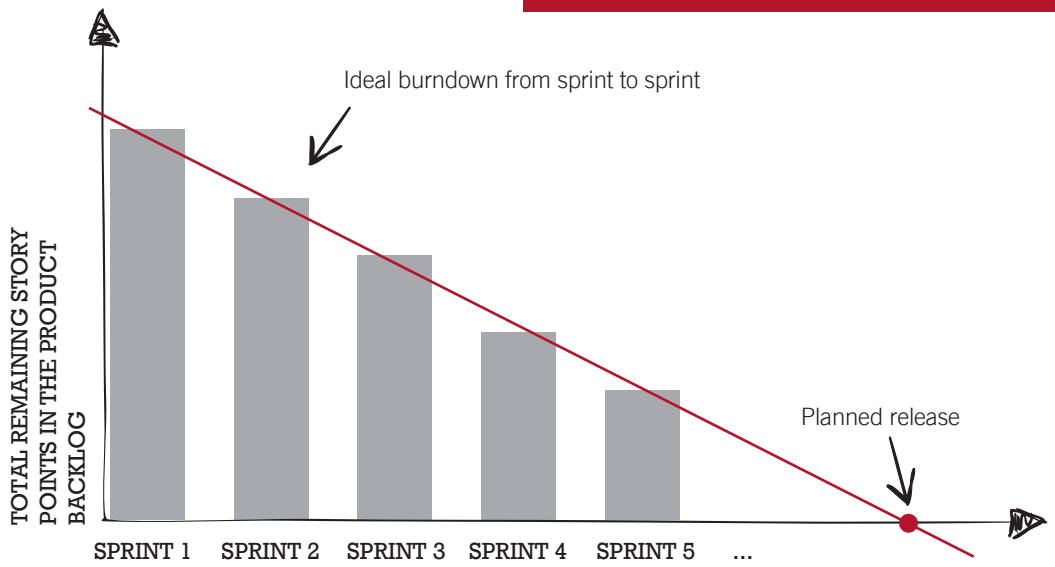
RELEASE FORECASTING

The release burndown chart can be used to ascertain the date of a release as soon as the Development Team's velocity is known or can be estimated and the total number of items in the product backlog for the release is stable.
An ideal burndown line can be plotted for the story points to be completed in a release and the velocity (number of story points expected to be completed in the sprint) showing the projected release date as the point of intersection between the ideal line and the x axis.

Sometimes the release date has to be brought forward or postponed due to changes in the product backlog or velocity over the course of the project.

Release burndown chart

- The chart is created on the basis of the user story estimates made at the outset of the project.
- Y axis: The total work to be completed in story points (total of all points for release-relevant user stories in the product backlog).
- X axis: Number of sprints.



PART 2 PEOPLE IN AGILE PROJECTS

SOFT FACTORS

Agile projects that use the Scrum method involve many different meetings, all of which have fixed structures.
The structures and content of these meetings are explained below. Then, the relevance of soft factors to the effective implementation and conclusion of team activities is described.

MEETINGS

IN AGILE PROEJCTS (SCRUM)

Meetings in Scrum projects take place in a fixed sequence. They are also **time boxed**, which makes them considerably more effective. Time boxed means that all meetings have a specific fixed duration and stop at the end of the timeframe.
If issues arise during a meeting that require further discussion, an extrameeting is scheduled and attended by the people affected by those issues. Strict rules apply at all meetings. Participants are expected to arrive punctually and comply with the code of conduct during the meeting (e.g. switch off mobile phones and focus on the topics being discussed).

TIME BOXED AND EFFICIENT SCRUM MEETINGS

Meetings	Max. duration	Frequency
Sprint Planning Meeting	2h for each weekly sprint	Once before the sprint
Daily Scrum	15 min.	Daily during the sprint
Sprint Review	1h for each weekly sprint	Once after every sprint
Sprint Retrospective	3 hours	Once after each sprint review

The backlog grooming meeting isn't an official Scrum meeting so it isn't time boxed.

PREPARING FOR A SPRINT

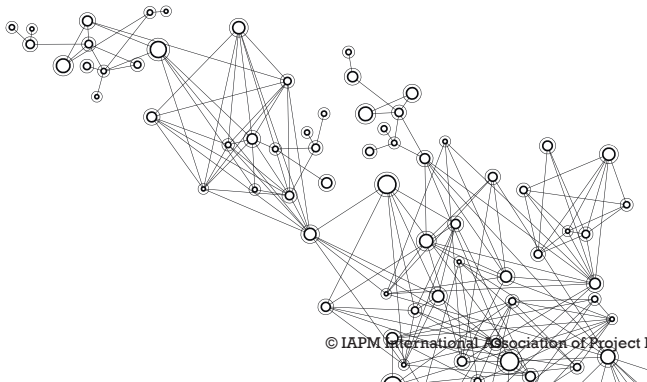
BACKLOG GROOMING OR BACKLOG REFINEMENT MEETING

When the team is preparing for the next sprint, it decides which of the items in the product backlog will be included in the sprint and what they require for their technical implementation. These preparations are generally done in the sprint planning meeting.

When the user stories are complex or the product backlog includes epics, it can be a good idea to hold an additional **backlog grooming meeting** to review and update the priorities in the product backlog, to divide up the product backlog items for the next sprint, if necessary, and to discuss them in detail. After the backlog grooming meeting, the Development Team can then focus on which user stories to include as tasks in the next sprint at the **sprint planning meeting**.

The backlog grooming meeting agenda could include the following items:

- Sorting and prioritisation of product backlog entries
- Deletion of obsolete product backlog entries
- Addition of new product backlog entries
- Detailed description of product backlog entries
- Grouping product backlog entries
- Estimating product backlog entries
- Release planning



SPRINT PLANNING

The Development Team focuses on two parameters when deciding how many user stories to select for implementation in the sprint at the sprint planning meeting:

The first is the net working hours available to the Development Team and the second is the estimated number of hours required to complete the tasks in the user stories.

The number of hours for each should be more or less identical if the team is to have a realistic chance of completing the tasks in the sprint on time.

In longer term projects, experience values and velocity can be used as the basis for selecting the user stories for each sprint.

The Product Owner ultimately decides which user stories are selected for each sprint.

At the end of the sprint planning meeting, the Development Team commits to processing the agreed number of user stories and delivering the product increment.

Sprint planning meeting

- Participants: Product Owner, possibly customer or user representative, Scrum Master, Development Team
- Duration: max. 8 hours for a 4-week sprint
- Facilitator: Scrum Master
- Responsible for goals: Product Owner
- product backlog items for the next sprint

DAILY SCRUM

THE 15-MINUTE STAND-UP MEETING

The daily scrum is a meeting that takes place every day during the sprint where the Development Team members can summarise their work progress.

Agenda

During the daily scrum, each team member answers the following three questions:

1. What did you do yesterday?
2. What will you do today?
3. Are there any impediments in your way?

Any impediments that are raised at the daily scrum meeting have to be resolved as quickly as possible by the Scrum Master. Sometimes he will create an impediment backlog to document the impediments and their elimination.

- Duration: 15 minutes, standing up
- Participants: Development Team and Scrum Master
- Ideally, the daily scrum meetings should be held in the morning. Attendance is mandatory and all Development Team members are expected to be punctual.
- Output: information from all Development Team members about the current status of their work and any impediments they are experiencing.

SPRINT RETROSPECTIVE

The sprint retrospective meeting is an opportunity for the Development Team to improve productivity and processes in the sprint. It is held after the sprint review. At the sprint retrospective, the team considers how impediment-free the processes were and what can be improved in future sprints.

- Participants: Scrum Master and Development Team
- Facilitator: Scrum Master
- Duration: max. 3 hours
- Output: review of the results of the previous sprint and information about 'how the sprint has gone' and 'where improvements can be made'.

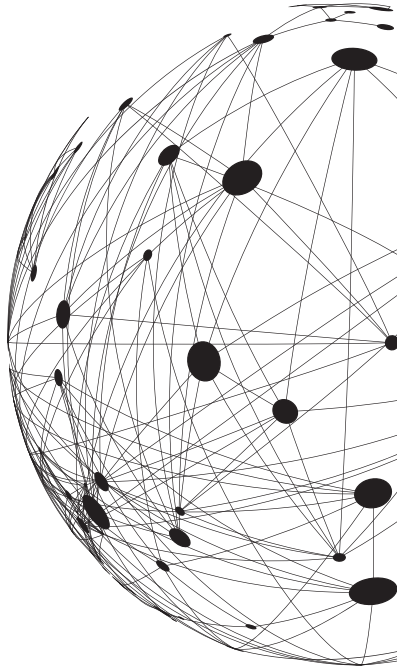
Here is a possible sprint retrospective agenda

- Participants' acceptance of the agenda (hand vote).
- Create a timeline for the previous sprint with post-it stickers.
- Mark specific items on the timeline with positive and delta cards.
- Discuss the results of the marked items.
- Identify necessary changes and improvements, or impediments to be eliminated in the next sprint (infrastructure, external interference, improvements in the Development Team etc.)
- Specification of action items, people responsible and completion deadlines.

TEAM BUILDING

The **GRPI model** is a process model that improves overall team development efficiency and team management.
The success of an agile project depends to a great extent on the participants' soft skills and the soft factors in the project. With the GRPI model, they can be used in a targeted way.

- G - Goals (definition and communication of goals)
- R - Roles (assignment and description of roles)
- P - Processes (what processes are taking place)
- I - Interpersonal (state of interpersonal relationships)



GOALS

GOAL DEVELOPMENT

Goals have to be defined and communicated. As soon as the product vision has been developed and the first version of the product backlog is ready, all the project goals and objectives should be communicated to the Development Team, environ-ment and stakeholders. Projects often veer off course because their participants don‘t all know what the plans are or the plans aren‘t communicated properly. That’s why both the scrum master and the Product Owner have to do everything possible in their roles to ensure that the whole Development Team is working towards the same goal.

ROLES

ROLE ASSIGNMENT

Roles are assigned and the content and responsibilities of each role are explained to the team members. In an agile project, the Development Team members decide among themselves which roles they will assume. If they cannot agree, the Scrum Master makes the decision.
The Product Owner liaises with stakeholders and markets the product after the project.

PROCESSES

PROCESS DEFINITION

It’s important to identify and define processes inherent to the Scrum framework. The agreed processes have to be adhered to so that the process model works.

This only works if processes are collectively agreed and communicated (see above).

INTERPERSONAL

RELATIONSHIPS

Interpersonal relationships are characterised by reciprocal communication, mutual understanding and empathy. The **sender-receiver principle** in communications theory tells us that all communications comprise messages being sent and received. This model also proposes that it is impossible not to communicate because breaking off communication or refusing to communicate also sends out a signal.

The following responsibilities apply when communicating information:

a) Sender’s responsibilities: ensure that the message has been properly received - by asking the recipient. Repeat if necessary.
b) Recipient’s responsibilities: ensure that everything has been correctly understood. Either repeat the message in own words or ask the sender what is meant to ensure that everything is understood.

A Scrum Master has to be able to recognise the phases that the Development Team goes through until it starts performing effectively. He also has to be able to identify problems and, if necessary, guide the team through each of the phases. Agile teams are **cross-functional,** which means their members represent all the key technical competences necessary to bring the project to a successful conclusion.

The original Development Team members should, if possible, remain in place throughout the project and replacements should be avoided. This is the key to a **performing team** that works at maximum **velocity** (see Part 1 of the Agile Guide).
The Scrum Master is a facilitator, while the Product Owner manages the stakeholders. If the agile project is a large-scale project, different Development Teams can be created, each headed by a different Product Owner, with each of these Product Owners reporting to a Chief Product Owner.

TEAM BUILDING PHASES

Step 1 - Forming
The cross-functional Development Team is formed by the Scrum Master with the approval of the base organisation and, if necessary, in collaboration with the Product Owner. Its members should have all the knowledge and skills necessary to implement the project. This is the phase where all team members meet at the outset of the project. They may know each other from working together in other projects or at the same organisation, or they may not know each other at all.

Step 2 - Storming
Now the Scrum Master has to prevent potential chaos by organising a team building workshop. This workshop provides a framework for discussing all the formalities of the agile project. However, it should also address social (informal) Development Team needs (getting to know each other, team development, appreciation of efforts etc.). Sometimes the Scrum Master has to be able to manage and eliminate conflicts in this phase - or act as mediator. Identification with the project and team building measures help to improve the team's day to day performance. Even so, conflicts can arise and have to be dealt with in the storming phase.

Step 3 - Norming
The frequency of Development Team conflicts usually abates in this phase and the members actually start to work together as a team. Individual team members have to get into a performing team as quickly as possible! Then the Development Team establishes an identity that is geared to the project.

Step 4 - Performing
In a performing team, all members can depend on each other, their activities and roles are clear and they need no discussion. In this phase, the team manages itself and the Scrum Master can take care of other things (such as promoting agile project management in the organisation or resolving conflicts with the line organisation). The team's velocity in sprints is relatively stable and predictable (completion of a certain number of story points in each sprint). Now it's important not to split the Development Team or swap existing members for new ones, otherwise it may be necessary to repeat earlier phases. All attempts by the line organisation to change the team's make-up are impediments that have to be dealt with by the Scrum Master.

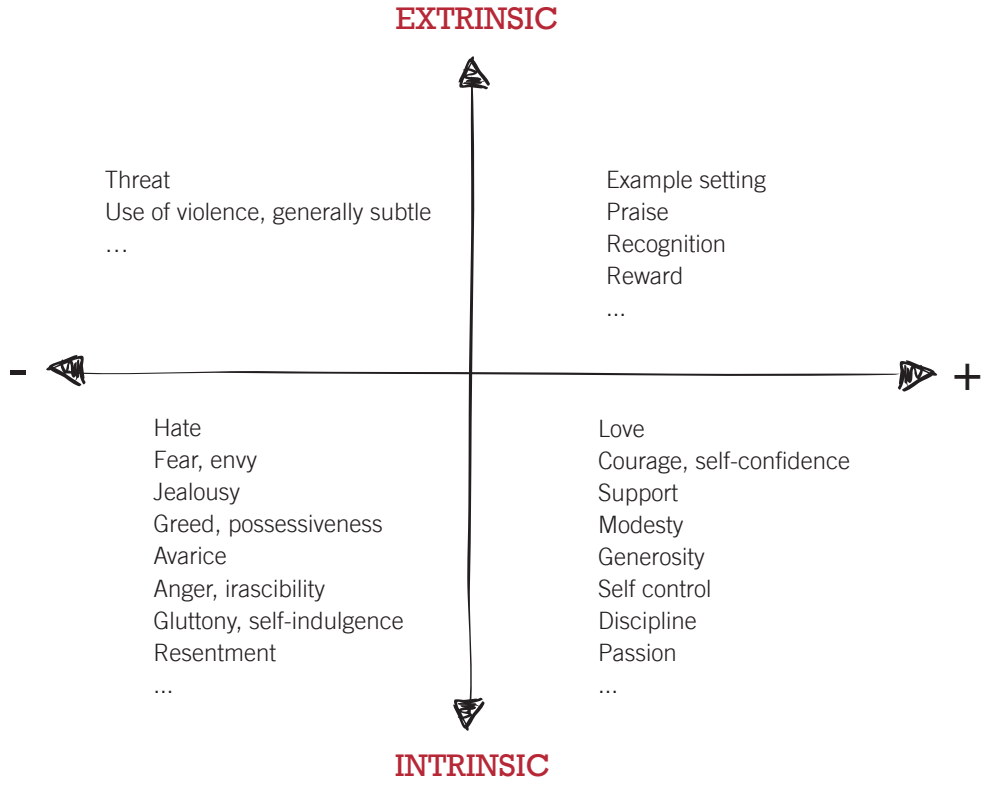
MOTIVATION

KNOWING THE MOTIVES

Project team members all have different professional qualifictions and different motivations. The success of a Scrum project depends to a great extent on how well the Development Team performs. This is the Scrum Master's responsibility.

Since he has no disciplinary authority over team members, the only motivation he can give them is positive. Negative motivation (threats, warnings etc.) can only be provided by the line managers.

A motivation matrix has to include the two basic types of motivation (intrinsic and extrinsic).



CONFLICT MANAGEMENT

All project participants (the Development Team, the Scrum Master and the Product Owner) should always try to prevent conflicts from arising in the first place through stakeholder management and adherence to Scrum rules. All meetings should be held at the scheduled time and for the scheduled duration. If conflicts do arise, they have to be dealt with as quickly as possible at the responsibility of all project participants. There are various conflict resolution options:

a) The people involved discuss the issue
b) The Scrum Master acts as a mediator
c) An (external) advisor is brought in
d) Escalation to a higher organisational level

The following checklist can be useful to the Scrum Master in conflict resolution:

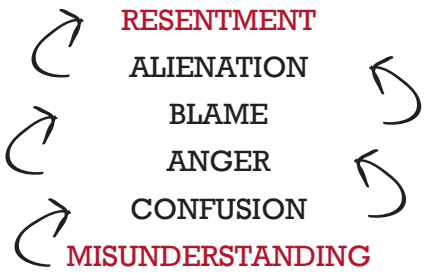
Have I

- identified the conflict partners correctly?
- prepared conflict partner profiles?
- put the conflict partners in an organisation chart?
- checked where the conflict is in the conflict spiral?
- applied the 10 points of conflict management?
- obtained professional advice (works council, company medical officer, company psychologist, mediator, arbitrator, legal expert, line manager, steering committee etc.)?
- reported, transferred, escalated the conflict to the proper persons or departments?

The four-point conflict resolution list below helps you to take a structured approach to dealing with a conflict and to find possible solutions.

- **Identify** the conflict
- **Address** the conflict
- **Find** a solution
- **Implement** the solution

THE CONFLICT SPIRAL



THE 10 POINT PLAN

1. Recognise the conflict.
2. Identify the people involved in the conflict.
3. Talk about the conflict.
4. Analyse the conflict.
5. Visualise the different sides of the conflict.
6. Categorise the conflict (relationship or emotional level?).
7. Take the conflict from the emotional to the objective level.
8. Structure the conflict.
9. Consider, evaluate, select and implement possible solutions.
10. Take advantage of the conflict to introduce new approaches.

PART 3 OTHER AGILE METHODS

KANBAN

Kanban is a scheduling system for just-in-time-production that was developed by Toyota and plays an integral role in the Toyota Production System (TPS). Kanban is a Japanese word which literally translated means **"card you can see or touch"**.

A **Kanban card** contains all the information that the employees need to know. This includes card type, Kanban ID, card number, item ID, item text, quantity, supplying source, source type, target destination, recipients, bar codes and product photo.

The objectives of Kanban are to improve productivity and quality and maximise production flexibility.

Achievement of these objectives prevents **wastage** and eliminates defects. Wastage in this sense refers to work which doesn't enhance the product's value, overwork of personnel and machines and process irregularities.

Surplus production is also wastage and should be avoided. It generates waste because the manufactured products which cannot be shipped take up storage space and tie up capital that the company then doesn't have available for other – and possibly more important – projects.

Kanban is essentially a **method for the procurement and supply of the necessary materials for each stage of production**. Kanban cards contain all information that is needed to obtain the required production items. They move between the supplier and customer processes.
Materials flows are all in a forward direction (from the supplier to the customer). **Information flows** are in a backward direction (from customer to supplier). That's why the Kanban method is also described as a **pull system**. A pull system is a process scheduling system that signals what to produce and which only produces the items that customers need. Pull systems are customer driven or demand oriented (i.e. demand for production orders comes from the end of the logistics chain).

KANBAN IN IT PROJECTS

Kanban was developed for production processes, which means it isn't suitable in its purest form for IT projects. Instead, it is combined with **lean production, lean development and the theory of constraints**.

Lean production is an aspect of lean management. It focuses on the cost-effective and time-efficient use of factors of production (people, machines etc.).

Lean development applies the lean production and lean management concepts to software development.

The **theory of constraints** is a management paradigm that views any system as being prevented from achieving more of its goals by a very small number of constraints. Improvements can only be made by identifying a constraint and restructuring the rest of the organisation around it.

There are no increments in Kanban. It's a continuous process.

The **workflow** is visualised and **work-in-progress** is limited.

VISUALISING THE WORKFLOW

The workflow is made visible to everyone involved in it on a **Kanban board** (e.g. a large whiteboard, or an electronic board). The board lists the individual process steps (e.g. requirement, design, implementation, test etc.) in columns.
The individual tasks are written on cards or post-it notes and stuck onto the board in lines (or above the relevant symbols in the electronic Kanban application). Often a combination of methods is used involving the Kanban board being "monitored" by a camera for changes.

The tasks/requirements can be formulated as **user stories, features, use cases** etc.

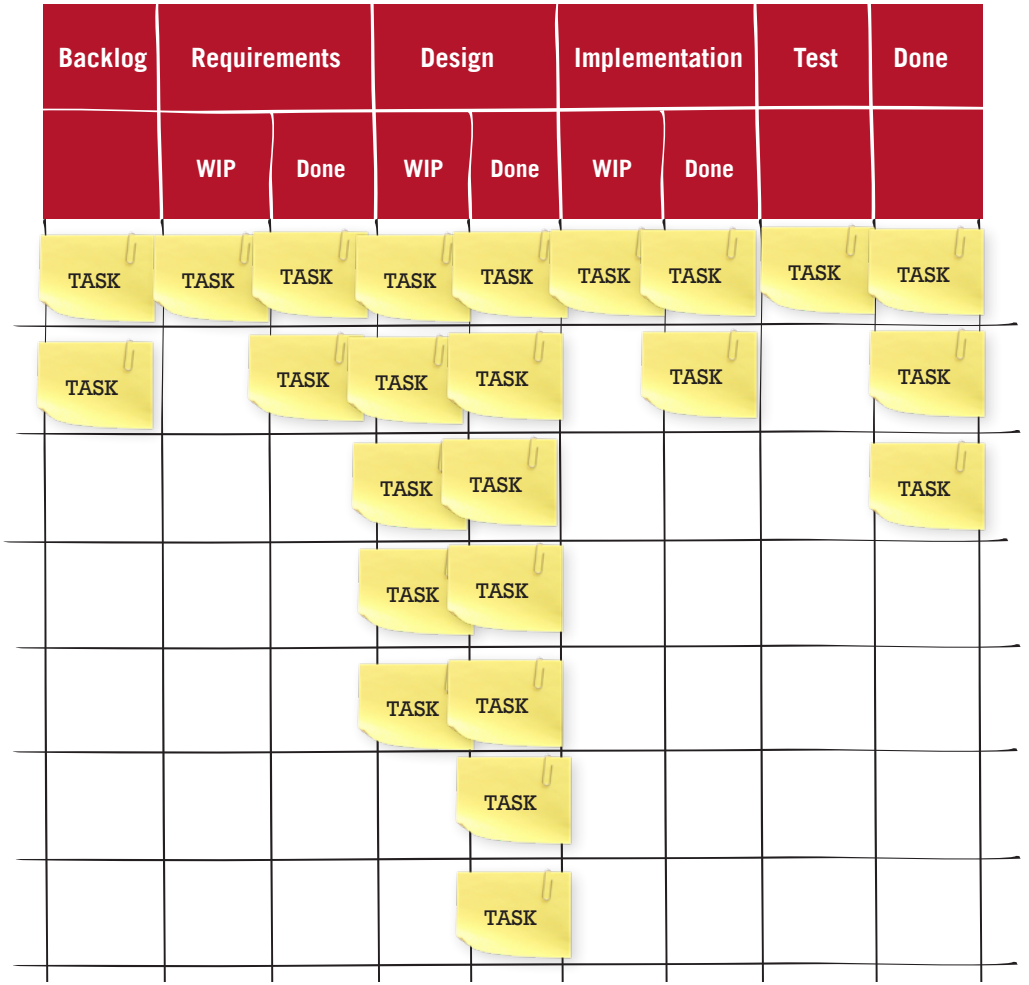
A **user story** describes what a user does or needs to do with the system in his or her job function.

A **feature** is a function (e.g. printout) in a software application.

Use cases describe the interaction between a the user and system to achieve a goal. A use case chart in **Unified Modeling Language** can also be useful.

Kanban teams use a pull system to optimise workflow from right to left across the Kanban board. The team pulls from the left when they have available capacity.

As a result, the cards/post-it notes on the Kanban board move from left to right until each task is completed.
There are no specific rules on what the Kanban board has to look like, so it is simply structured according to individual team requirements.

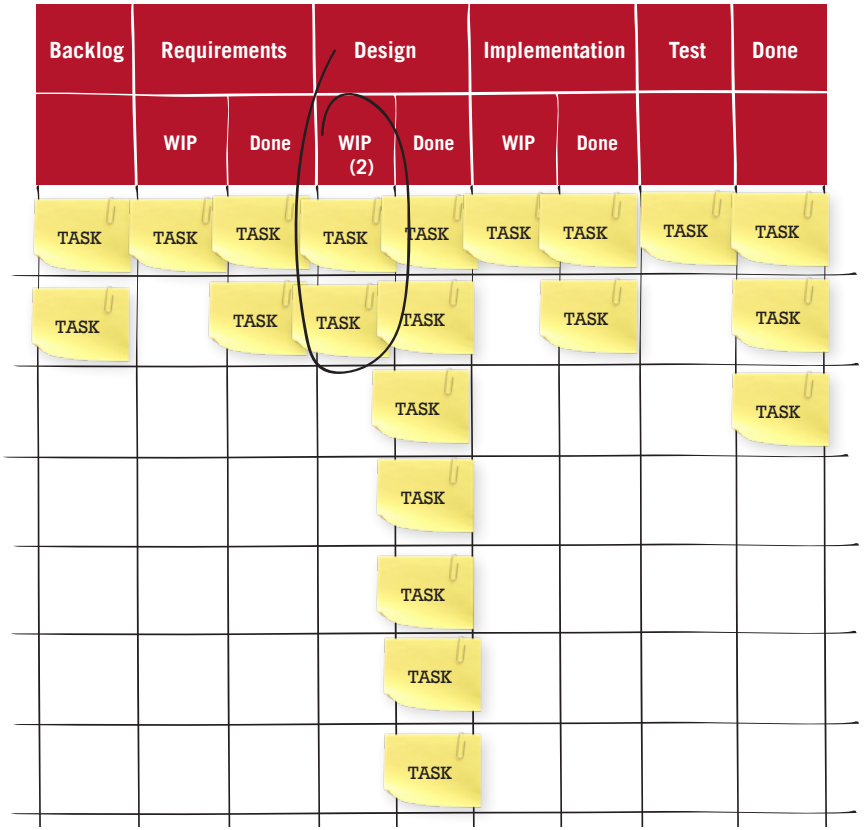


LIMITING WORK IN PROGRESS

When a Kanban board is used, the entire team is aware of the status of tasks and able to identify constraints.

One example of an implementation constraint is when the design features can't all be implemented at the same speed. This makes the workflow uneven.

Limiting the amount of **design work** in progress can remedy the situation. Limiting design WIP means that only two cards can be included in the WIP column on the Kanban board. This ensures that work in progress can be completed before new work is started.



SCRUM VS.KANBAN

A note in advance: Kanban has less rules than Scrum.

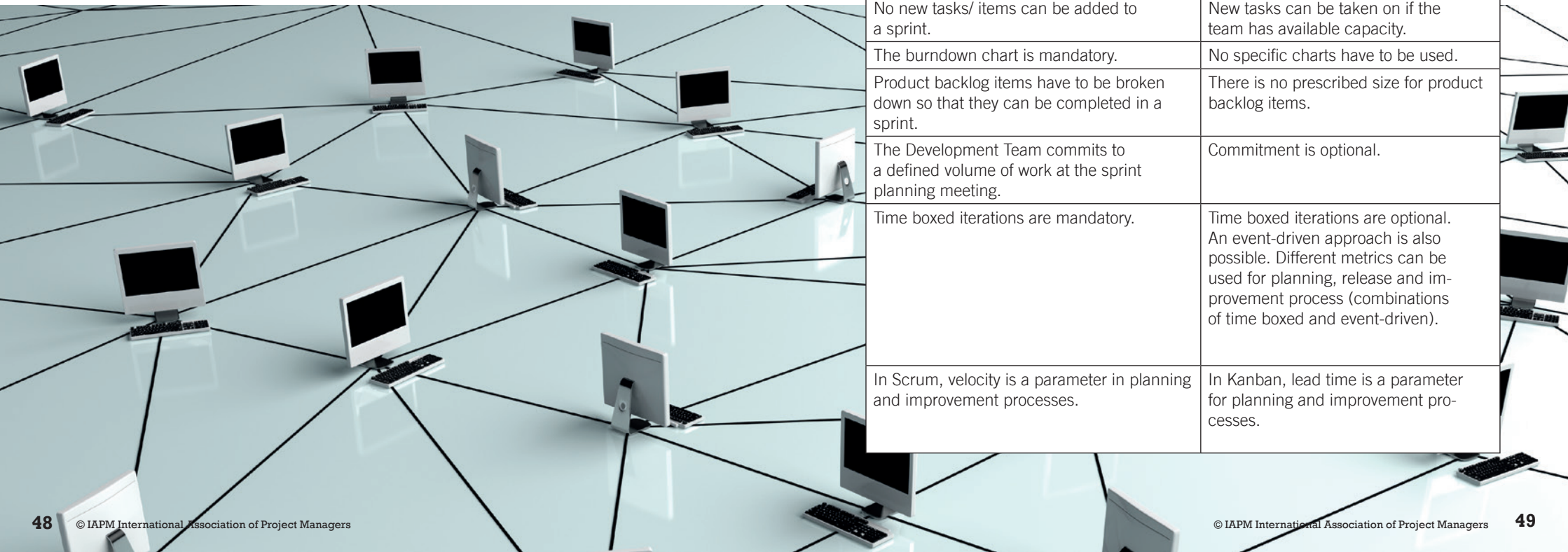
SIMILARITIES

Scrum and Kanban...

- improve processes through transparency and workflow visualisation,
- are both agile methods,
- speed up and increase the frequency of release-capable software components,
- are pull systems,
- limit concurrent work in progress,
- have self organising teams,
- requirements are broken down into detailed steps and
- they have few rules.

DIFFERENCES

Scrum	Kanban
There are three roles in Scrum (Product Owner, Scrum Master and Development Team).	Kanban has no defined roles. If necessary, roles can be defined in specific projects by the participants.
WIP is indirectly limited (in each sprint).	WIP is directly limited (in each process step).
Estimating is mandatory.	Estimating is optional.
The Scrum task board is cleared after every sprint.	The Kanban board is continuously updated.
The backlog is prioritised.	Prioritisation of requirements is optional.
The sprint backlog is Development Team-specific.	The Kanban board can be used by several teams/individuals.
No new tasks/ items can be added to a sprint.	New tasks can be taken on if the team has available capacity.
The burndown chart is mandatory.	No specific charts have to be used.
Product backlog items have to be broken down so that they can be completed in a sprint.	There is no prescribed size for product backlog items.
The Development Team commits to a defined volume of work at the sprint planning meeting.	Commitment is optional.
Time boxed iterations are mandatory.	Time boxed iterations are optional. An event-driven approach is also possible. Different metrics can be used for planning, release and improvement process (combinations of time boxed and event-driven).
In Scrum, velocity is a parameter in planning and improvement processes.	In Kanban, lead time is a parameter for planning and improvement processes.



EXTREME PROGRAMMING (XP)

Extreme programming is a discipline of **agile** software development that accords low significance to formal processes. It takes an incremental approach to achieving customer requirements and was developed in the Chrysler Comprehensive Compensation System project as an agile model for software development. It is based on values of simplicity, communication, feedback, courage and respect. In addition to these **values**, XP also has specific **principles** and **practices**. Like Scrum, there are also roles in XP. The main ones are customer, product owner (internal, often project manager) and development team.

VALUES

Communication
All team members should communicate verbally whenever possible. As a result, questions can be answered faster and more directly, and misunderstandings cleared up quicker.

Courage
Focusing on the values and practicing open and direct communication embody courage. Continuous adaptation to changes and the reduction of customer requirements to the essentials can be challenging to people who aren't accustomed to upholding these values in projects.

Feedback
Fast, direct and continuous customer feedback improves the quality of the product. It ensures that flaws in the system are quickly recognised and recoded so that customer gets the product he needs.

Respect
The customer should respect the developers' know-how and vice-versa. Every member of the project team deserves to be respected and should, in turn, show respect.

Simplicity
Simple designs can be implemented faster than complex designs and they are often less expensive. That's why XP encourages starting with the simplest solution.

PRINCIPLES

The principles in XP are based on the values just described, and are intended to be more easily translated to guidance in a concrete XP situation.

Accepted responsibility
Responsibility is not delegated or assigned but accepted. As a result, each individual identifies more strongly with his or her function.

Assume simplicity
Simple designs can be implemented faster, at lower cost and are easier to understand and maintain. The lower the degree of complexity of a software, the easier it is to provide feedback.

Concrete experiments
Abstract decisions should be supported by a series of experiments to reduce potential risks.

Embrace change
The project team should unreservedly embrace change.

Honest measurement
Measurements are necessary to steer the project and gauge its progress. These measurements should be transparent, honest and comprehensible to all project participants.

Incremental change
Changes should be implemented incrementally, or in small steps. Then, they are easier to understand, less complex and involve less dependencies than complex changes.

Local adaption
XP is adapted to local needs and requirements.

Open, honest communication
Project team members embrace honest, open and direct communication.

Play to win
These XP principles are designed to facilitate the success of the project. The developers don't simply want to produce perfect code, but a harmonious software package. Although this is a goal in other agile methods, XP in particular goes beyond the code.

Quality work
User feedback has a big impact on product quality. Objective (negative and positive) feedback is an important contributor to development team satisfaction. The development team should always be able to work in a framework that permits quality work.

Rapid feedback
The time between the activity and feedback on that activity is crucial. Feedback is another explicit element of quality assurance.

Small initial investment
By focussing on the important functions, the team can quickly develop the first viable prototypes.

Teach learning
Project team members have to learn what they need to achieve their goals, e.g. which and how many software tests the developers have to perform.

Travel light
The XP team has to get used to travelling light, which means using just a few tools, resources and methods. All they need are the tools that produce value for the project.

Work with people's instincts, not against them
Trust the team's instincts, even when it chooses an unconventional approach to achieving its goal. (Refer to the section on values and "respect".)

PRACTICES

The values and principles are supplemented with practices. These practices help the developers to follow the principles.

MANAGEMENT PRACTICES

On-site customer
A customer representative should be on-site to clarify requirements and questions directly. This function is often performed by the project manager.

Planning game
Before each increment the content of the next increment is discussed and planned by all project team members, including the customer or its representative (internal project manager).

Short releases
The development cycles for new releases should be short, e.g. daily. This gives the customer fast access to new or changed functions. Also, the customer can provide prompt feedback to the development team so that they are quickly aware of any flaws in the system.

TEAM PRACTICES

Coding standards
Any developer can work on any piece of code. Team standards for coding should be established so that the entire team can share responsibility for the code.

Continuous integration
To avoid excessive dependencies and provide the customer with timely feedback, the change integration process is continuous.

Collective code ownership
Any developer can work on any piece of code at any time. This means that all the programmers get to see all the parts of the code. They are collectively responsible for the code as a team.

System metaphor
The system metaphor is a story that everyone can tell about how the system works. All project team members should be clear about how the system works.

Sustainable pace
The concept is that programmers shouldn't work more than a 40 hour week, and if there is overtime in one week, the next week should not include more overtime. People perform best and most creatively when they are rested.

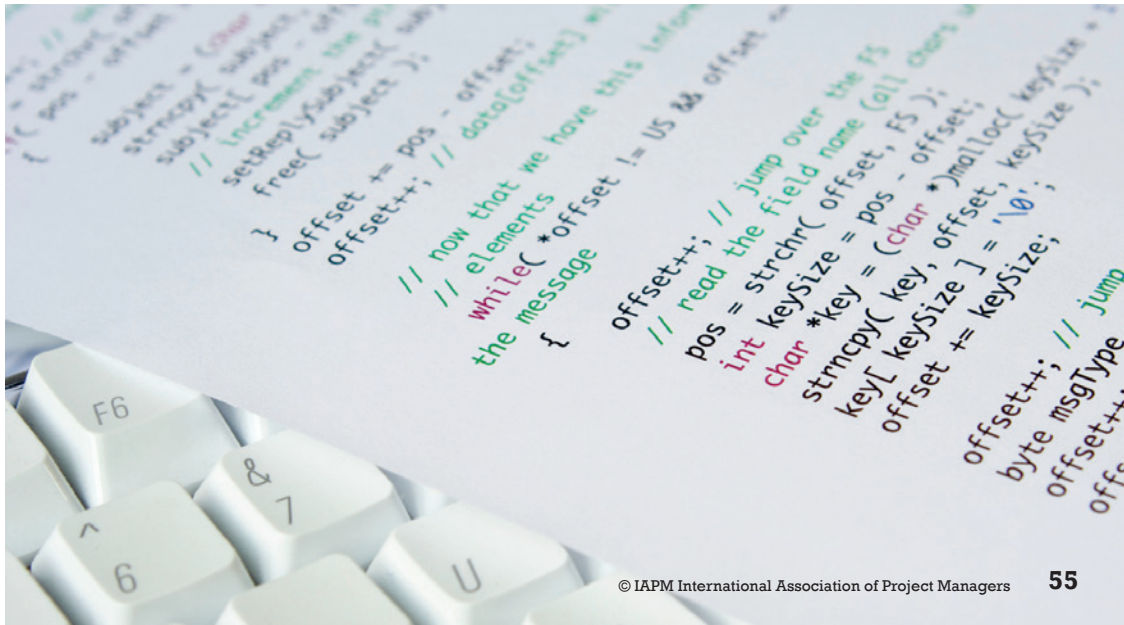
PROGRAMMING PRACTICES

Pair programming
All code is produced by two people programming on one task on one workstation. One programmer has control over the workstation and is thinking mostly about the coding in detail. The other programmer is more focused on the big picture, and is continually reviewing the code that is being produced by the first programmer. This approach helps to prevent source code errors and flaws in the system.

Refactoring
Refactoring is the process of restructuring the source code without changing its behaviour to improve non-functional attributes or reduce its complexity. Component tests prevent the occurrence of side effects.

Simple design
The less complex the system is, the easier, faster and less expensive it is to implement. Other advantages are that the system is easier to maintain, understand, test and upgrade.

Testing
Automated component tests are performed on the source code to validate it. The user checks whether requirements have been met in acceptance tests.



SCRUM VS. EXTREME PROGRAMMING

Scrum is an agile project management method that doesn't dictate how a software project is implemented.
XP is an agile software development method that covers the entire project lifecycle.

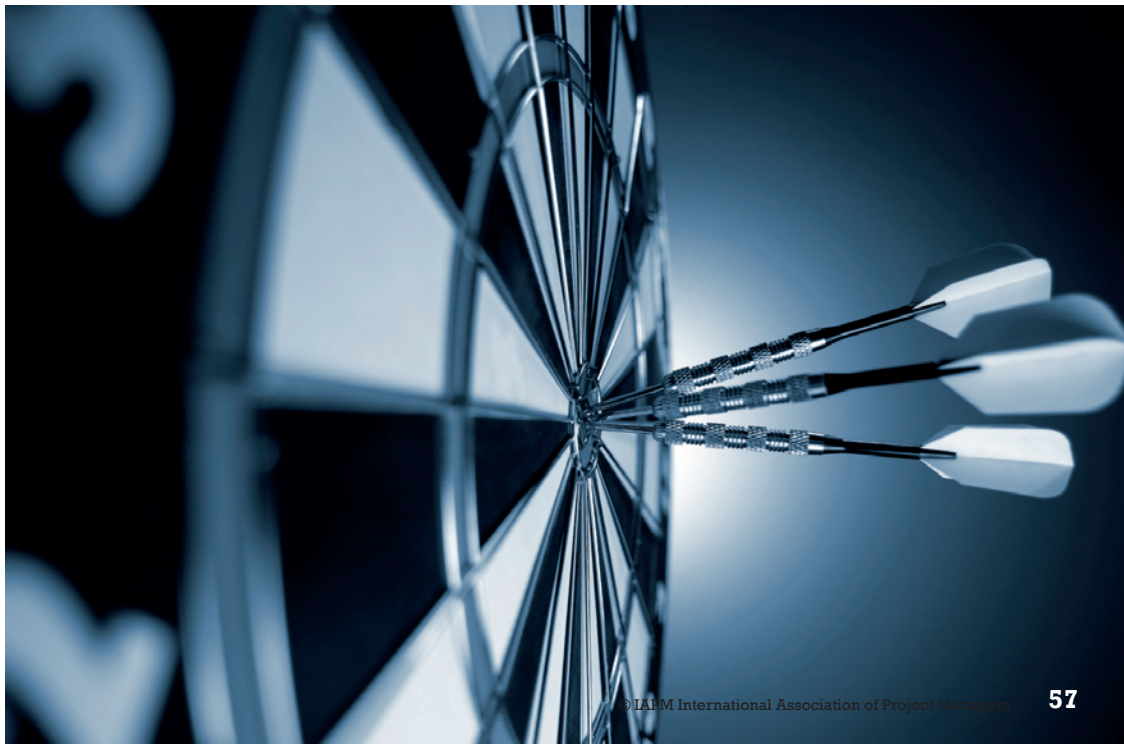
Scrum	Extreme Programming
Development Teams typically work in iterations called sprints that are between two and four weeks in length.	Extreme Programming iterations are generally one to two weeks in length.
Requirements do not change during a sprint.	If the XP team hasn't yet started working on a specific feature, it can swap it for another feature of the same size.
Scrum does not dictate how the project is to be implemented.	Extreme Programming involves Test-Driven Development, automated tests, pair programming, refactoring, etc.

SCRUM AND EXTREME PROGRAMMING

Scrum and Extreme programming are often confused with each other because they have several philosophical similarities.

- People and communication take precedence over processes and tools.
- Operable software is more important than detailed documentation.
- Close cooperation with the customer is preferred to lengthy contract negotiations.
- Embracing change is more important than strict adherence to the plan.

A combination of the two methods can be practical because they share a similar philosophy and complement each other.



PART 4
THE IAPM-CERTIFIED AGILE PROJECT MANAGER

INTRODUCTION
CERTIFIED AGILE PROJECT MANAGER (IAPM)

The IAPM certification process is based on the IAPM Agile Project Management Guide in its valid version. The basis for the Agile Project Management Guide is the agile knowledge base of IAPM. The knowledge base was developed in co-operation with an international panel of experts and managers in an agile environment. It is continuously adapted to international standards and findings.

The Agile Project Management Guide is available free of charge on the IAPM website (www.iapm.net).

Whether the certificate holder has acquired his knowledge from the Agile Project Management Guide or from other documents is irrelevant for admission to the examination. The IAPM supports the independent preparation for the certification exams.

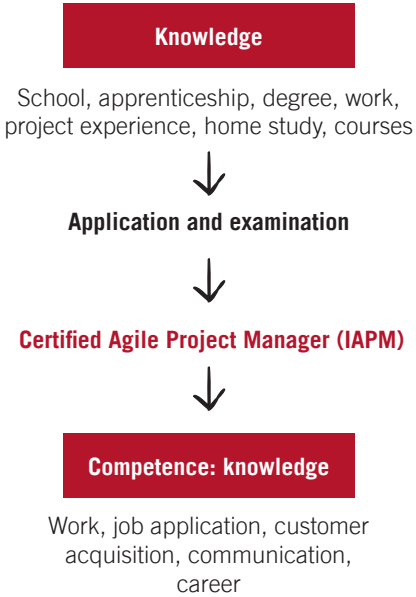
THE CERTIFICATIONS

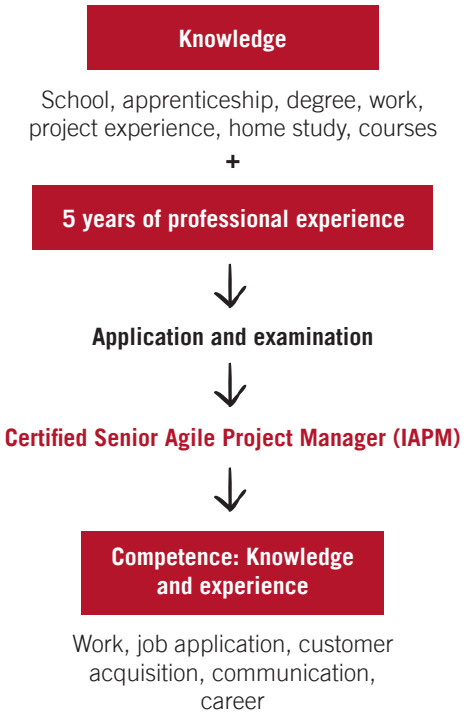
REQUIREMENTS

Certified Agile Project Manager (IAPM) certification is intended for managers who want to obtain confirmation of their agile project management knowledge. Candidates have to demonstrate a knowledge of all Scrum competence elements in the **online examination.** Previous experience isn't necessary, but it is advantageous.

Certified Senior Agile Project Manager (IAPM) certification is geared to project managers with at least five years of verified agile project management experience, three of those as project manager or in a senior mana-gement role. Candidates for both certifications have to take an **online examination** and provide proof of practical experience.

The Agile Project Management Guide forms the basis for the knowledge elements, in which among other things the hard and soft factors of agile project management are presented.





PROCEDURE

APPLICATIONS AND ADMISSIONS

REGISTRATION

To register for certification, you have to fill in an **application form.**

It is an **online** form and can be found at www.iapm.net. When the form has been filled in and all the necessary information has been provided, the documents are checked by the IAPM for completeness and compliance with admission requirements.

If all the requirements are met, the IAPM issues an invoice to you.

When your payment is received, you are sent your login data for the **online test or examination** (either the self test or certification examination).

TEST AND EXAMINATION

SELF TEST AND CERTIFICATION EXAMINATION

The IAPM offers two options – **an online self test and a certification examination** – for Certified Agile and Certified Senior Agile Project Manager (IAPM) certification. They can be taken on any PC, laptop or tablet.

SELF TEST AND CERTIFICATION EXAMINATION

If you are not sure whether you can pass the certification examination without a preparatory course, you can take a **self-test** on the IAPM website. This provides an initial idea of the types of questions and level of difficulty of the certification examination.

Our system assesses whether you have passed the self test and provides you with e-mail feedback on the areas of agile project management where you have knowledge gaps so that you can revise these areas more thoroughly.
If you decide that you would like to improve your knowledge in a preparatory course, you can register for a German or English language course with any of the IAPM Training Partners. Take a look at our website to find a trainer near to you.

To pass the **certification examination** a minimum score of 65% has to be achieved. Examination result notifications are sent out by e-mail.

CERTIFICATES

When you have passed the certification examination your certificate will be sent to you by e-mail. Each certificate has a unique identification code (IAPMIC) which has to be validated on the IAPM website, stating first name and surname.

FAILED THE CERTIFICATION EXAMINATION?

If you don't manage to pass the examination you can repeat it with a different set of questions. In this case, you have to pay the examination fee again. If you also fail this second examination, you have to wait for 12 months before you can have another attempt. If you fail for a third time, you are excluded from any further IAPM certification examinations and are not able to retake the examination again.

AFFIRMATION IN LIEU OF OATH

Certification candidates are required to provide the following affirmation in lieu of oath before they can take the examination. The affirmation in lieu of oath is archived in the certification candidate's personal file. This information is treated as strictly confidential and it is stored in accordance with the IAPM's privacy policy and the European Union's data protection legislation.

1. I agree to satisfy and conduct myself in accordance with all IAPM certification programme policies and requirements.

2. I shall maintain confidentiality of IAPM examination questions and content. Furthermore, I agree not to copy, discuss, debrief or disclose, in any manner, the specific content of IAPM examination questions and answers to any individual.

3. I certify and swear that I have performed the exam entirely alone and without the help of any other individual, literature or any other means of assistance.

4. I agree that I shall at all times act in a truthful and honest manner and provide truthful and accurate information to the IAPM. I agree that any intentional or unintentional failure to provide true, timely and complete responses to questions in this application may lead to further investigation and/or sanctions by the IAPM.

5. I agree that all materials that I submit to the IAPM certification department become the property of the IAPM certification department, and that the IAPM certification department is not required to return any of these materials to me.

6. I agree that information related to my participation in the IAPM certification process may be used in an anonymous manner for research purposes only.

7. I agree that all disputes relating in any way to my application for an IAPM certificate and/or my involvement generally in an IAPM certification program, will be resolved solely and exclusively by means of IAPM certification department policies, procedures and rules. Any disputes or lawsuits are excluded.

8. The IAPM reserves the right to suspend or revoke the certificate of any individual who is determined to have failed to uphold, or otherwise breached this agreement. In this case, the IAPM reserves the right to pursue criminal proceedings.

9. I release and indemnify the IAPM and the IAPM certification department from all liability and claims that may arise out of, or be related to, my project management and related activities.

10. I hereby release, discharge and indemnify the IAPM, its directors, officers, members, examiners, employees, attorneys, representatives, agents and the IAPM certification department from any actions, suits, obligations, damages, claims or demands arising of or in connection with this application, the scores given with respect to the examination or any other action taken by the IAPM with regard to certifying, testing and professional development including, but not limited to, all actions related to ethics matters and cases.

11. I understand and agree that any decision concerning my qualification for any certificate, as well as any decisions regarding my continuing qualification for any certificate rest within the sole and exclusive discretion of the IAPM, and that these decisions are final.

IMPRESSUM

www.iapm.net

2nd edition

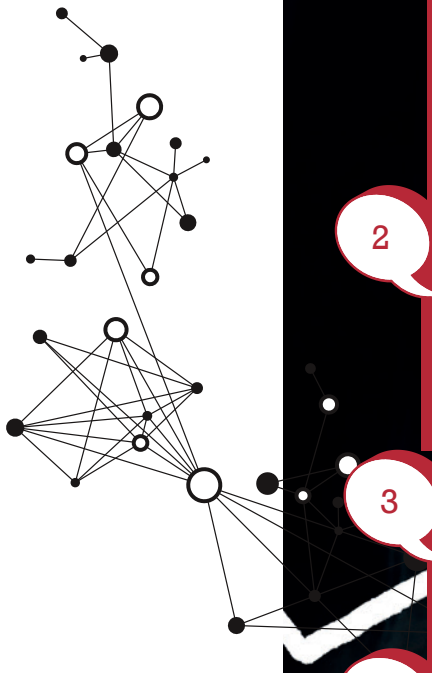
Photography: www.istockphoto.com

Edited by IAPM International Association of Project Managers™ in Liechtenstein

ISBN: 978-3-941739-28-4

Quality Management System
The quality management system used by the IAPM International Association of Project Managers ™ satisfies the requirements of ISO 9001.

Trademark Protection
IAPM International Association of Project Managers ™ is a protected EU trademark – no. 9539354 –



THE BENEFITS

HOW YOU CAN BENEFIT FROM IAPM CERTIFICATION

1 Competitive advantages & career launching pad

- Proven agile project management competence
- Competitive advantages for organisations and individuals
- Standardisation of terms and methods with the Agile PM Guide 2.0
- External, objective verification of knowledge and experience

2 Online examinations

- No travel expenses
- No pressure of time to prepare
- Exams can be taken on any PC

3 No re-certification necessary

- No certificate expiry date
- No new costs

4 Fair fees

- The fees depend on the GDP of the country in which the certificate candidate has citizenship.

5 Anonymous Certification

- No subjective evaluations
- No "fail quota"
- No discrimination