Cairo University
Faculty of Graduate Studies for Statistical Research
Department of Computer and Information Sciences

Software Development Process

Lec. 2 outline: You’ll find all the information you need here on Roles, Responsibilities, and Methodologies, The Team Around the Product, Roles and Responsibilities, Methodologies: Waterfall, Methodologies: Agile..., and so on..

Dr. Mohamed Fakhry: Consultant in Advanced AI Engineering, Accounting Solutions, and Software Development

E-mail: 12422021625503@pg.cu.edu.eg; mfdeveloper22@gmail.com Tel: 01201208623

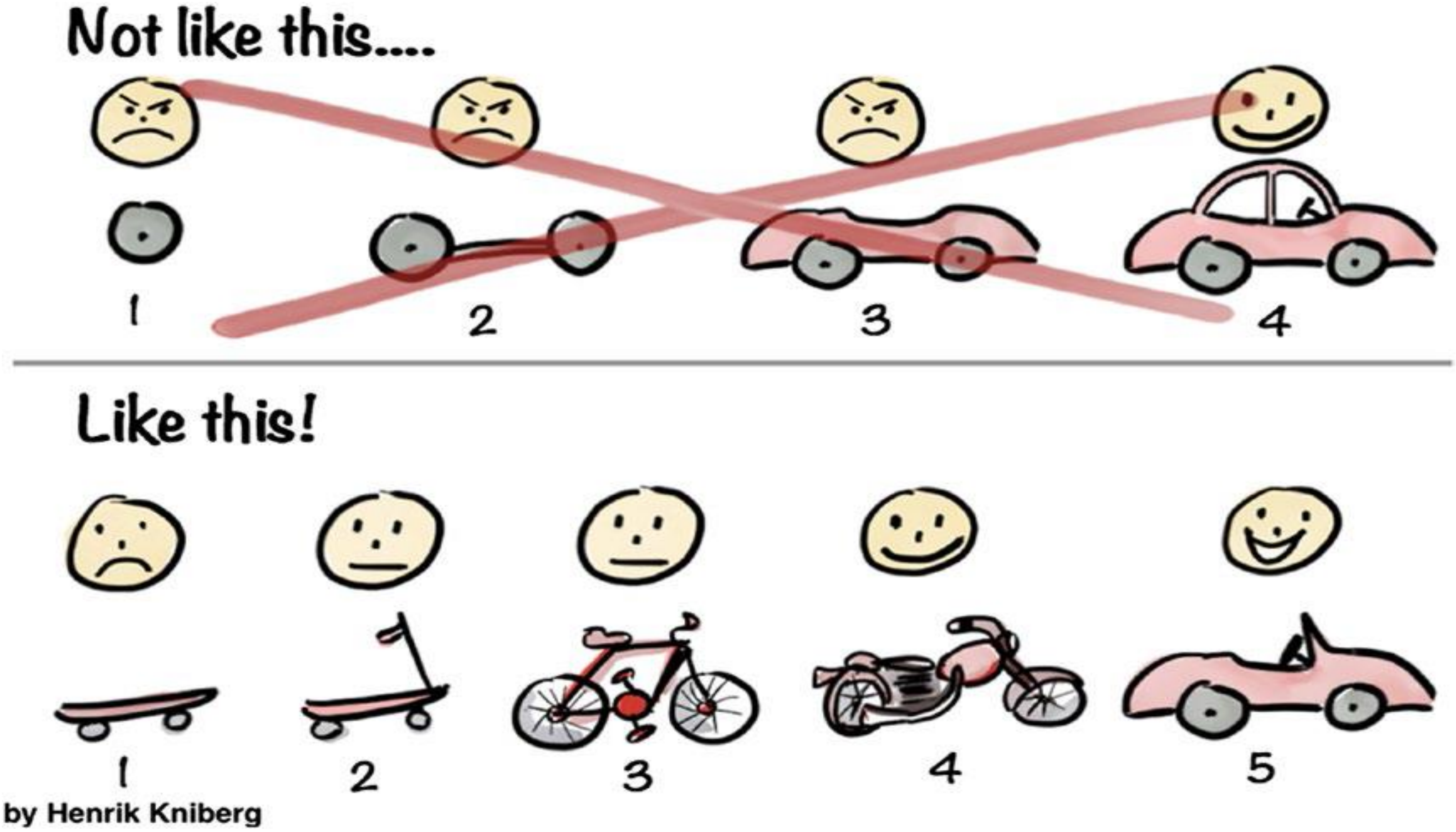
References

- □ Software Development From A to Z: A Deep Dive into all the Roles Involved in the Creation of Software, Olga Filipova and Rui Vilao , 2018
- □ ISBN-13 (pbk): 978-1-4842-3944-5 ISBN-13 (electronic): 978-1-4842-3945-2
- □ https://doi.org/10.1007/978-1-4842-3945-2

Course notes

Web support articles and cases study

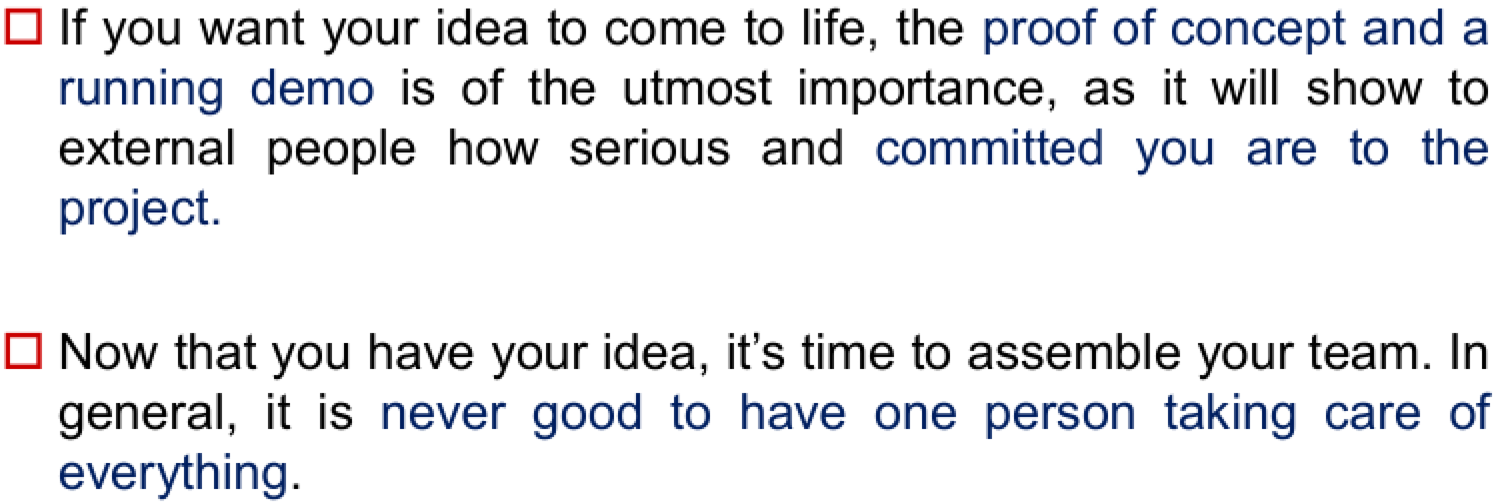
Famous illustration by Henrik Kniberg about MVP



Agenda

- Roles, Responsibilities, and Methodologies
- The Team Around the Product
- Roles and Responsibilities
- Methodologies: Waterfall
- Methodologies: Agile
- Agile: Scrum
- Agile: Kanban

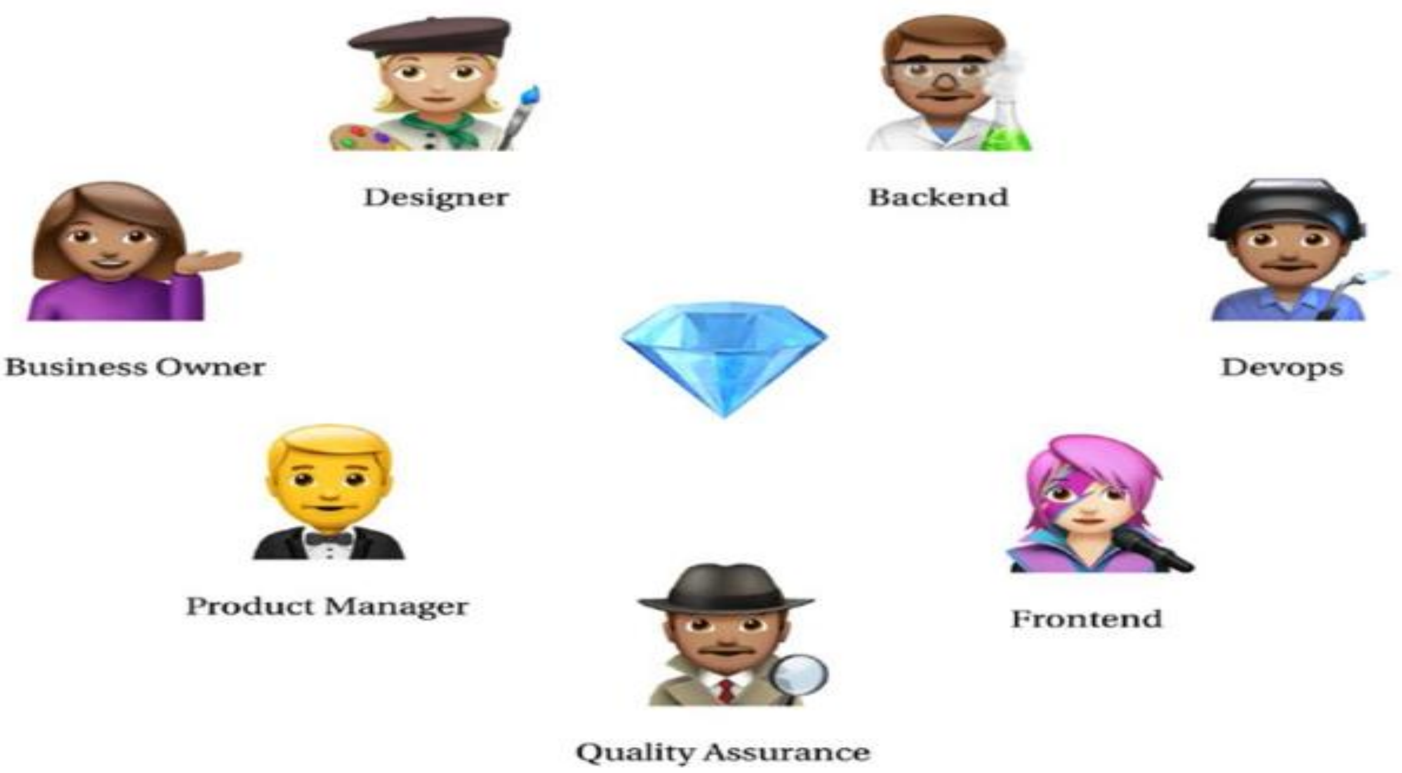
Roles, Responsibilities, and Methodologies



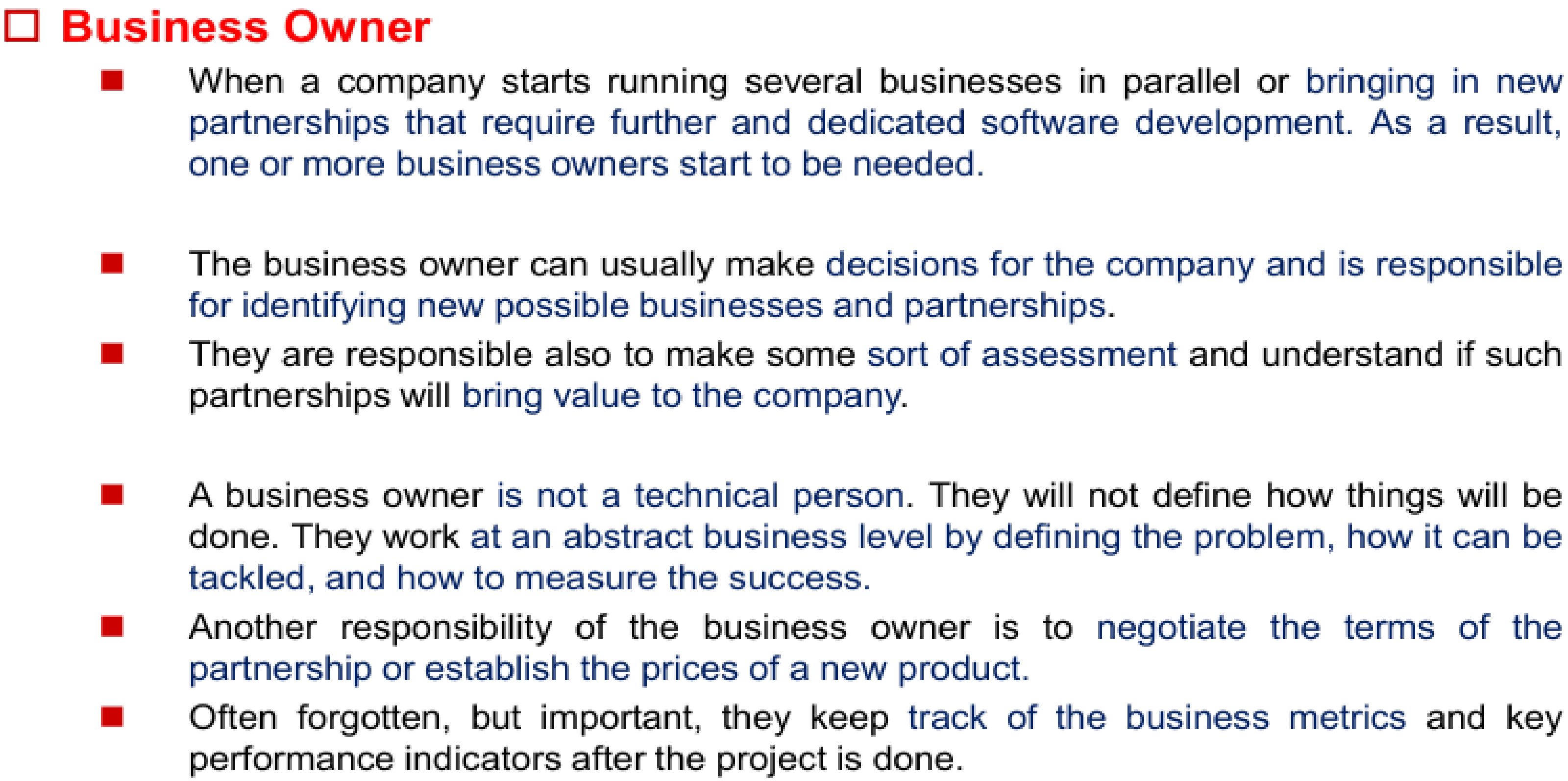
- □ If you want your idea to come to life, the proof of concept and a running demo is of the utmost importance, as it will show to external people how serious and committed you are to the project.

- □ Now that you have your idea, it's time to assemble your team. In general, it is never good to have one person taking care of everything.

The Team Around the Product



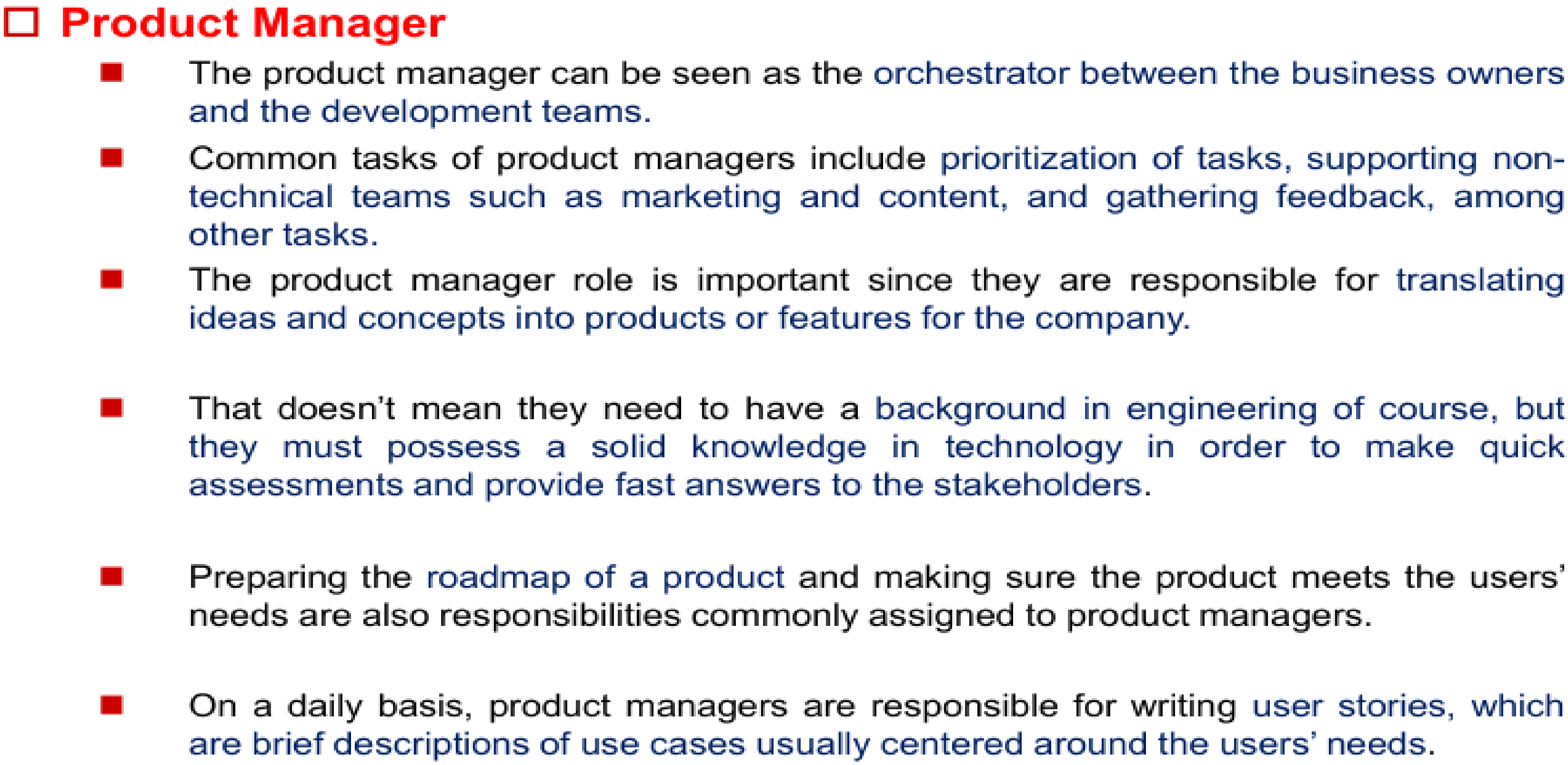
Roles and Responsibilities



□ Business Owner

- When a company starts running several businesses in parallel or bringing in new partnerships that require further and dedicated software development. As a result, one or more business owners start to be needed.
- The business owner can usually make decisions for the company and is responsible for identifying new possible businesses and partnerships.
- They are responsible also to make some sort of assessment and understand if such partnerships will bring value to the company.
- A business owner is not a technical person. They will not define how things will be done. They work at an abstract business level by defining the problem, how it can be tackled, and how to measure the success.
- Another responsibility of the business owner is to negotiate the terms of the partnership or establish the prices of a new product.
- Often forgotten, but important, they keep track of the business metrics and key performance indicators after the project is done.

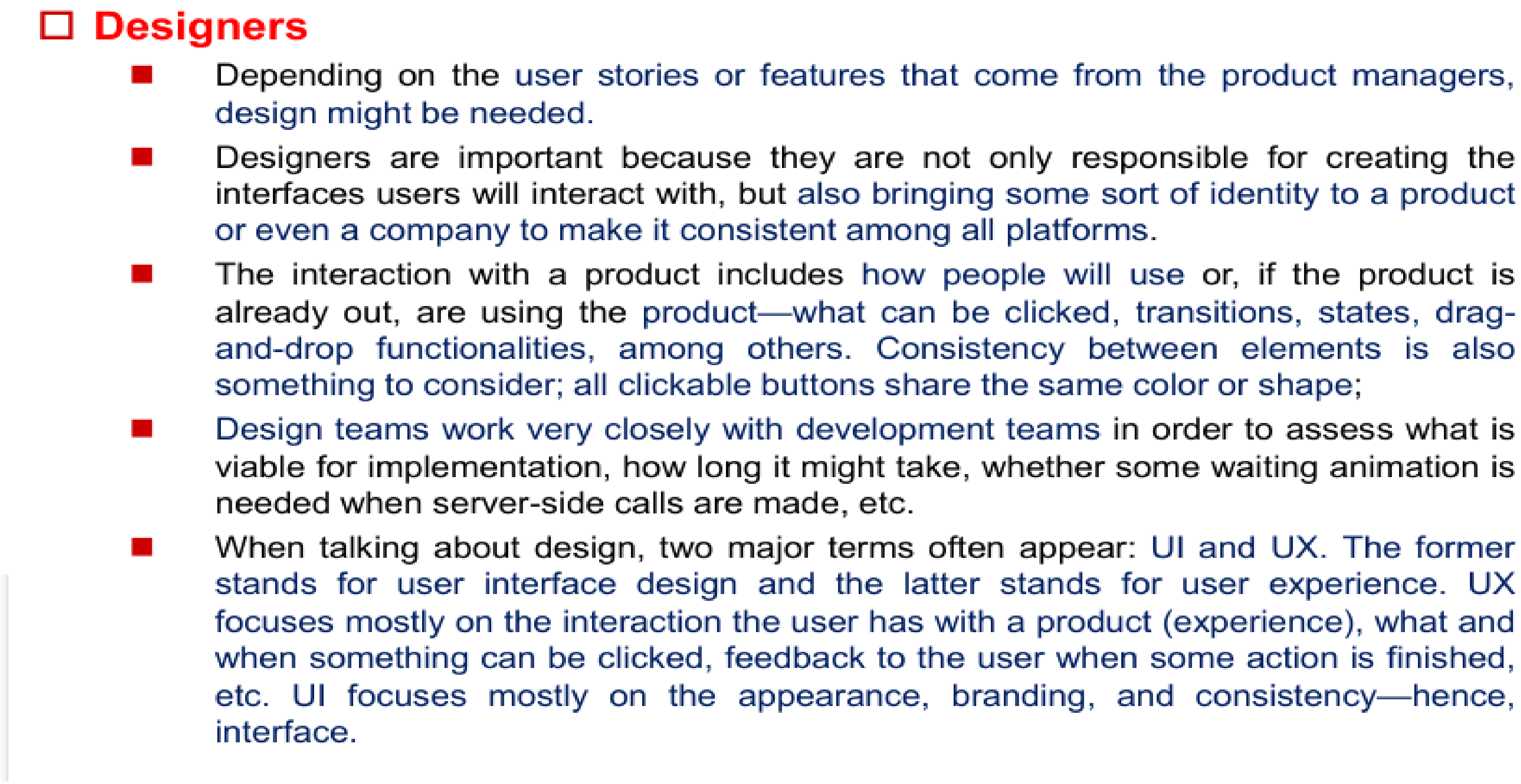
Roles and Responsibilities Cont.



□ **Product Manager**

- The product manager can be seen as the orchestrator between the business owners and the development teams.
- Common tasks of product managers include prioritization of tasks, supporting non-technical teams such as marketing and content, and gathering feedback, among other tasks.
- The product manager role is important since they are responsible for translating ideas and concepts into products or features for the company.
- That doesn't mean they need to have a background in engineering of course, but they must possess a solid knowledge in technology in order to make quick assessments and provide fast answers to the stakeholders.
- Preparing the roadmap of a product and making sure the product meets the users' needs are also responsibilities commonly assigned to product managers.
- On a daily basis, product managers are responsible for writing user stories, which are brief descriptions of use cases usually centered around the users' needs.

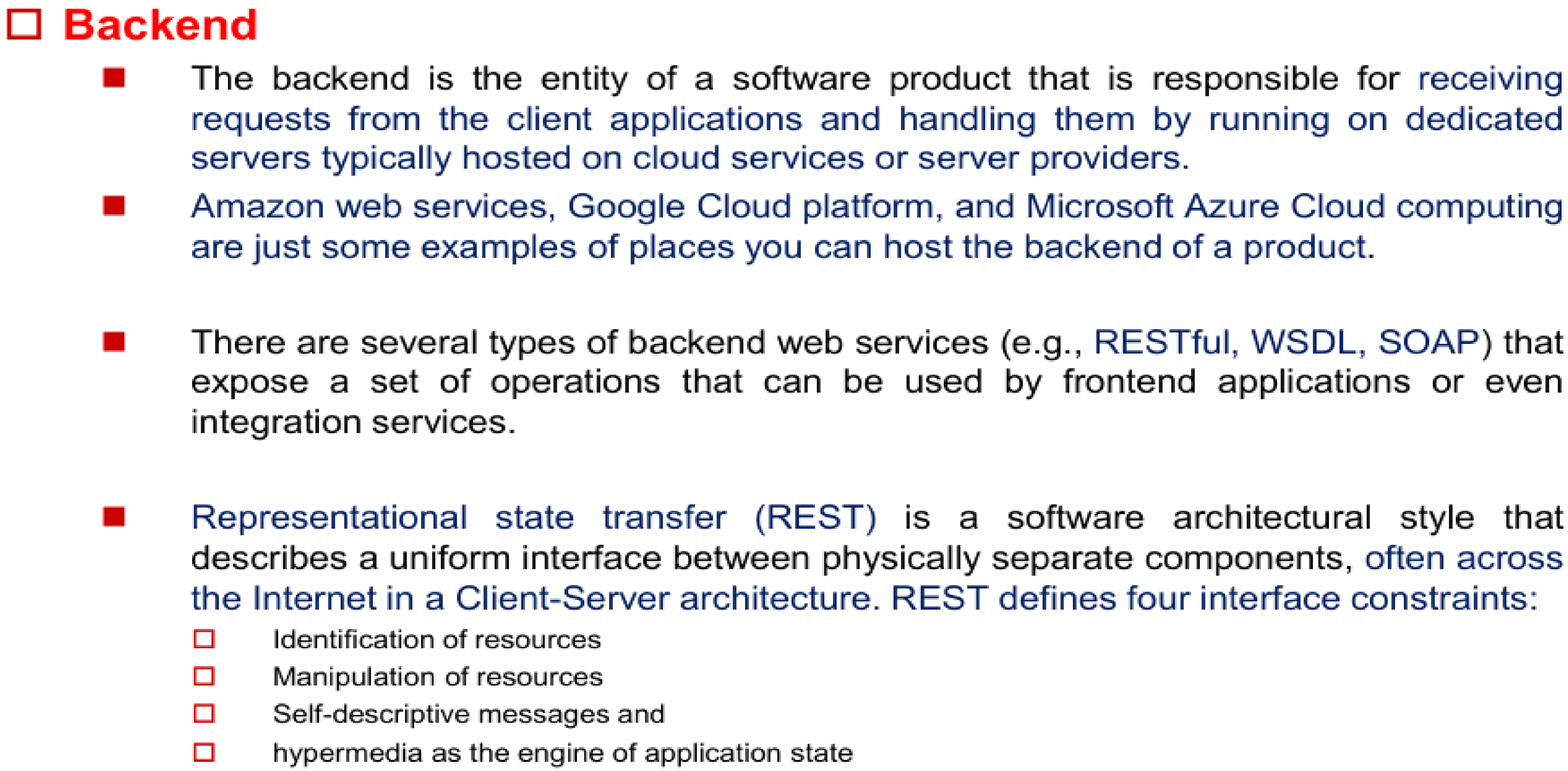
Roles and Responsibilities Cont.



□ **Designers**

- Depending on the user stories or features that come from the product managers, design might be needed.
- Designers are important because they are not only responsible for creating the interfaces users will interact with, but also bringing some sort of identity to a product or even a company to make it consistent among all platforms.
- The interaction with a product includes how people will use or, if the product is already out, are using the product—what can be clicked, transitions, states, drag-and-drop functionalities, among others. Consistency between elements is also something to consider; all clickable buttons share the same color or shape;
- Design teams work very closely with development teams in order to assess what is viable for implementation, how long it might take, whether some waiting animation is needed when server-side calls are made, etc.
- When talking about design, two major terms often appear: UI and UX. The former stands for user interface design and the latter stands for user experience. UX focuses mostly on the interaction the user has with a product (experience), what and when something can be clicked, feedback to the user when some action is finished, etc. UI focuses mostly on the appearance, branding, and consistency—hence, interface.

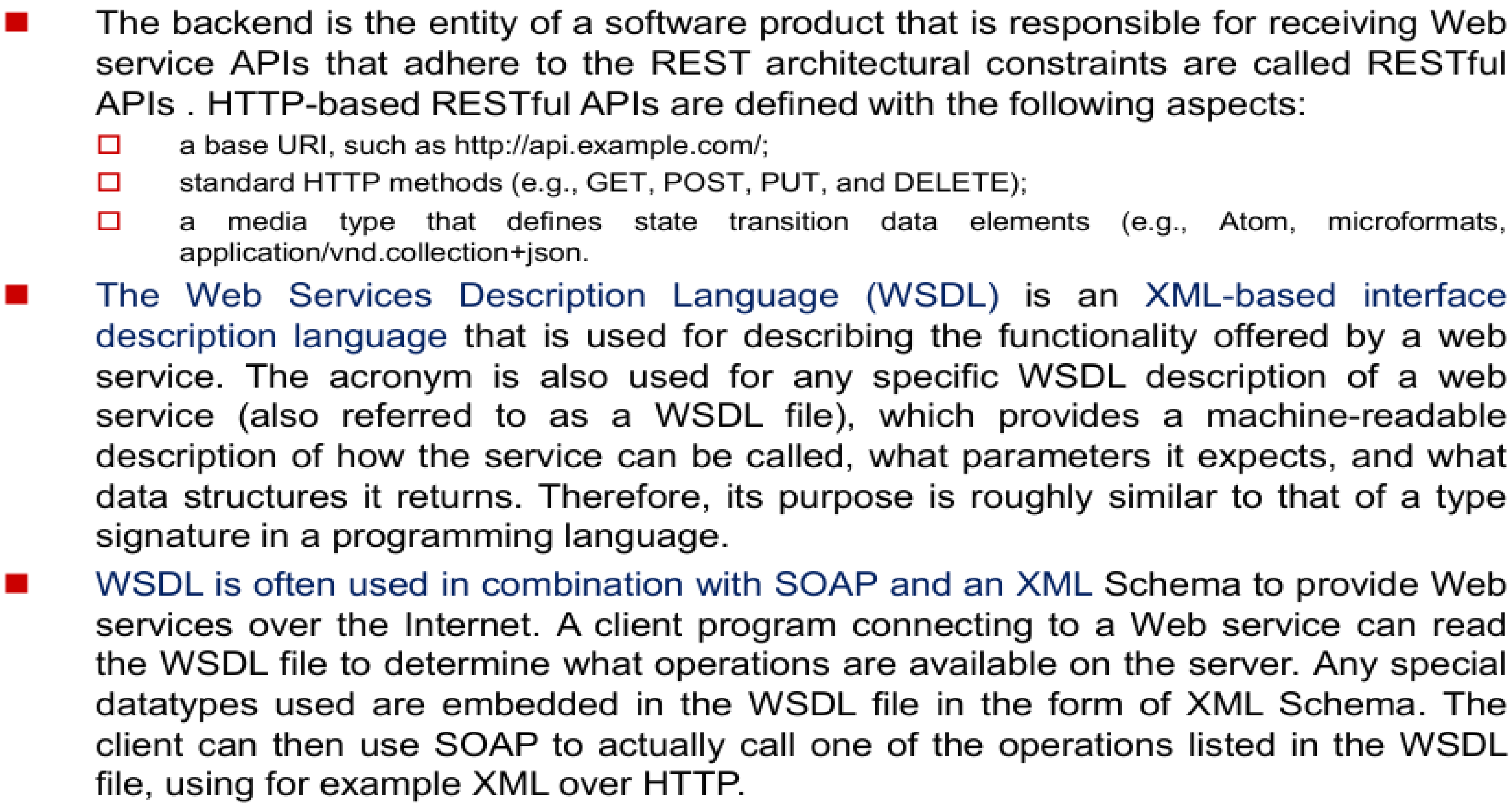
Roles and Responsibilities Cont.



□ Backend

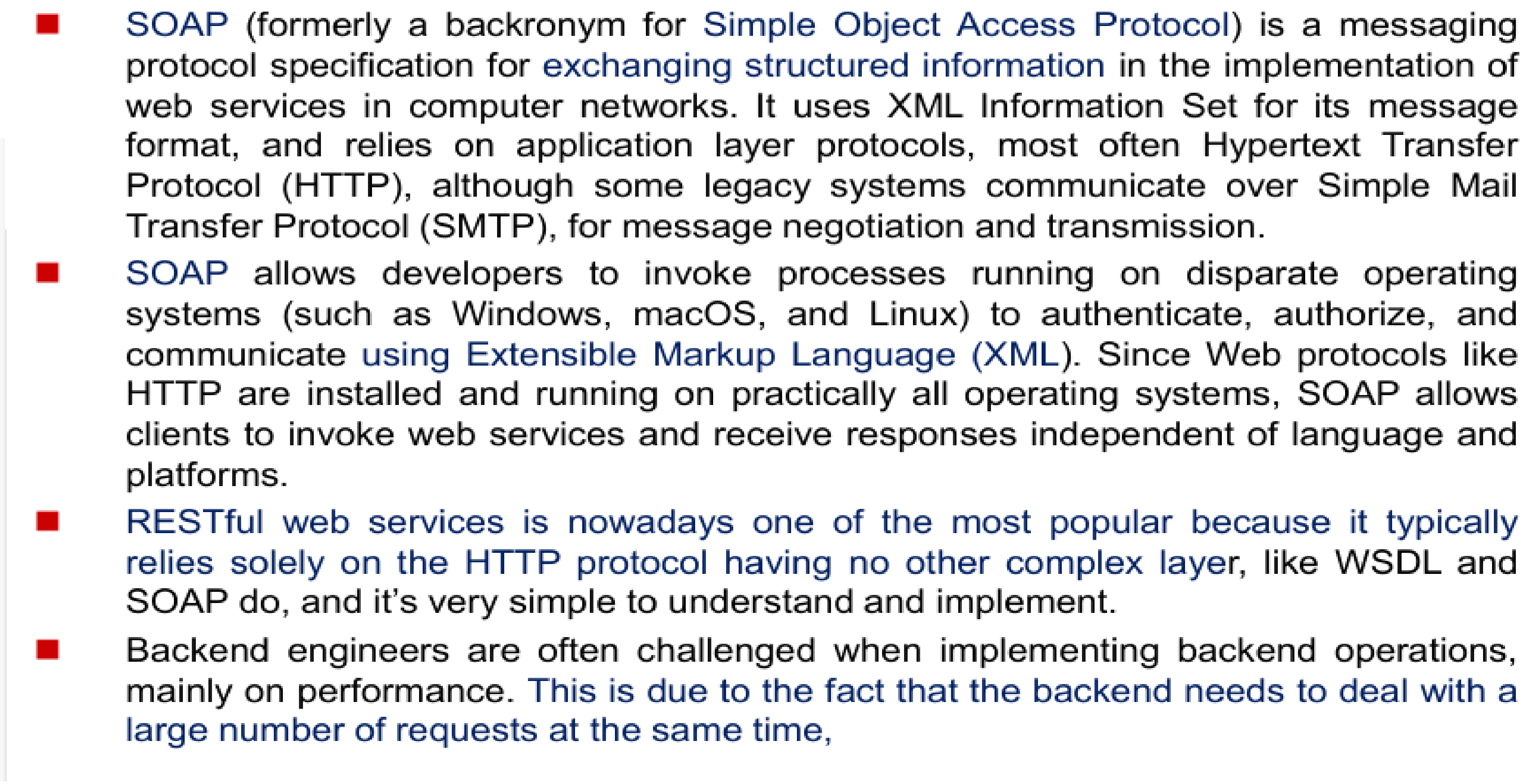
- The backend is the entity of a software product that is responsible for receiving requests from the client applications and handling them by running on dedicated servers typically hosted on cloud services or server providers.
- Amazon web services, Google Cloud platform, and Microsoft Azure Cloud computing are just some examples of places you can host the backend of a product.
- There are several types of backend web services (e.g., RESTful, WSDL, SOAP) that expose a set of operations that can be used by frontend applications or even integration services.
- Representational state transfer (REST) is a software architectural style that describes a uniform interface between physically separate components, often across the Internet in a Client-Server architecture. REST defines four interface constraints:
  - □ Identification of resources
  - □ Manipulation of resources
  - □ Self-descriptive messages and
  - □ hypermedia as the engine of application state

Roles and Responsibilities Cont.



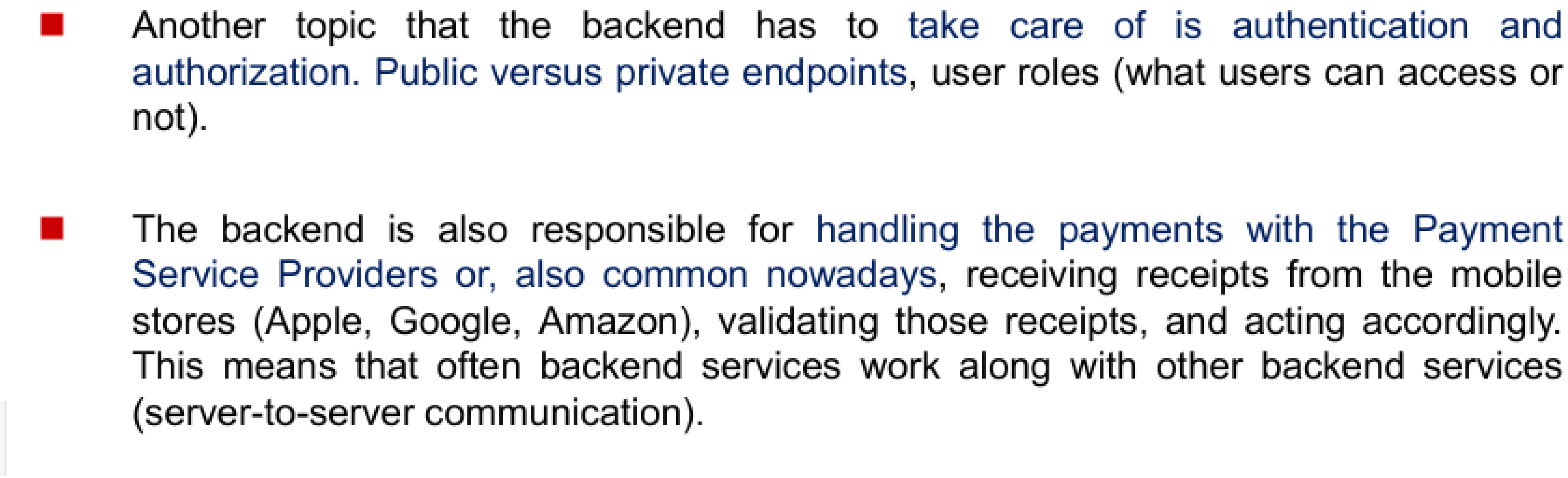
- The backend is the entity of a software product that is responsible for receiving Web service APIs that adhere to the REST architectural constraints are called RESTful APIs . HTTP-based RESTful APIs are defined with the following aspects:
  - □ a base URI, such as http://api.example.com/;
  - □ standard HTTP methods (e.g., GET, POST, PUT, and DELETE);
  - □ a media type that defines state transition data elements (e.g., Atom, microformats, application/vnd.collection+json.
- The Web Services Description Language (WSDL) is an XML-based interface description language that is used for describing the functionality offered by a web service. The acronym is also used for any specific WSDL description of a web service (also referred to as a WSDL file), which provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns. Therefore, its purpose is roughly similar to that of a type signature in a programming language.
- WSDL is often used in combination with SOAP and an XML Schema to provide Web services over the Internet. A client program connecting to a Web service can read the WSDL file to determine what operations are available on the server. Any special datatypes used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the operations listed in the WSDL file, using for example XML over HTTP.

Roles and Responsibilities Cont.



- SOAP (formerly a backronym for Simple Object Access Protocol) is a messaging protocol specification for exchanging structured information in the implementation of web services in computer networks. It uses XML Information Set for its message format, and relies on application layer protocols, most often Hypertext Transfer Protocol (HTTP), although some legacy systems communicate over Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.
- SOAP allows developers to invoke processes running on disparate operating systems (such as Windows, macOS, and Linux) to authenticate, authorize, and communicate using Extensible Markup Language (XML). Since Web protocols like HTTP are installed and running on practically all operating systems, SOAP allows clients to invoke web services and receive responses independent of language and platforms.
- RESTful web services is nowadays one of the most popular because it typically relies solely on the HTTP protocol having no other complex layer, like WSDL and SOAP do, and it's very simple to understand and implement.
- Backend engineers are often challenged when implementing backend operations, mainly on performance. This is due to the fact that the backend needs to deal with a large number of requests at the same time,

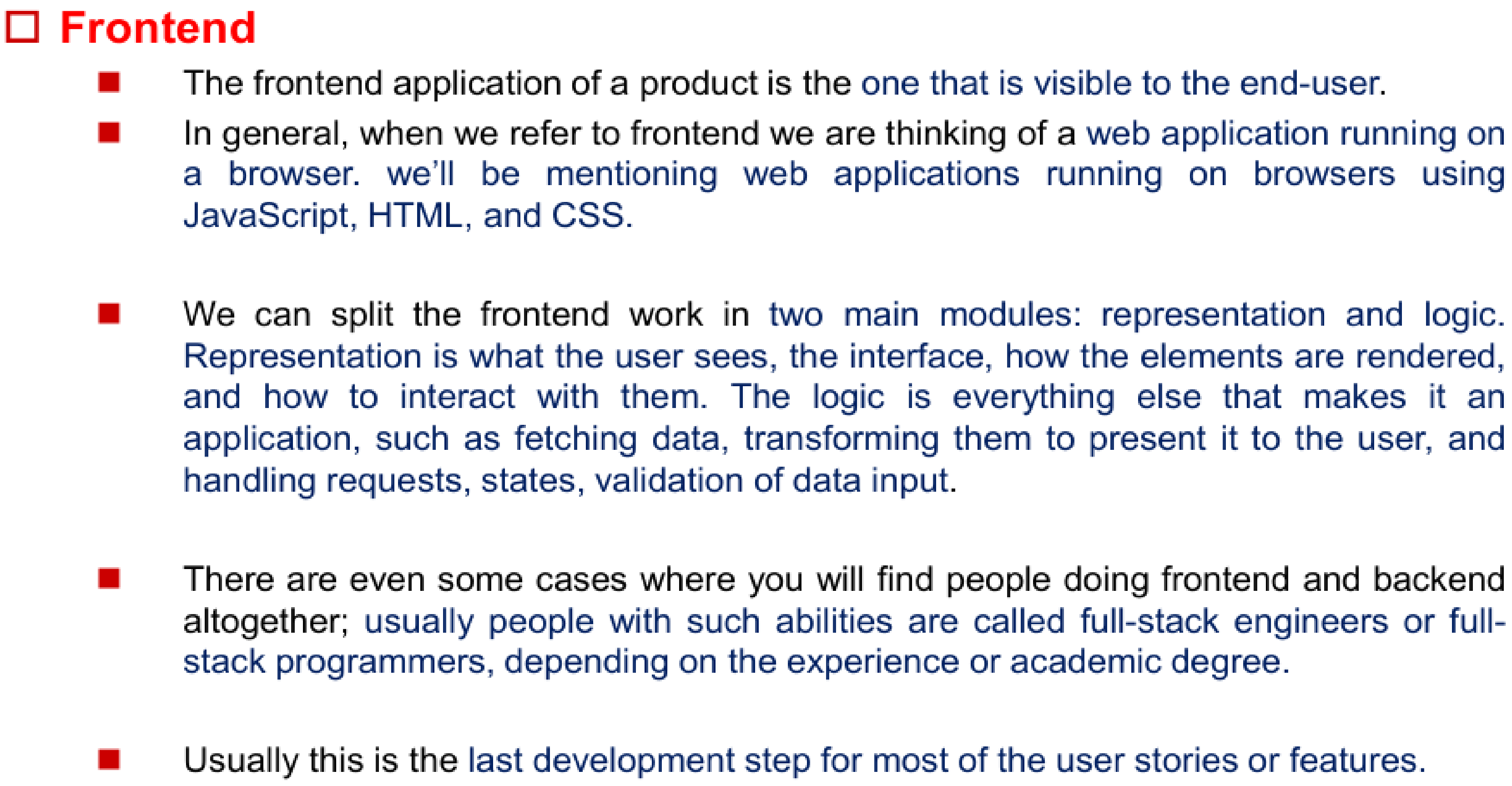
Roles and Responsibilities Cont.



- Another topic that the backend has to take care of is authentication and authorization. Public versus private endpoints, user roles (what users can access or not).

- The backend is also responsible for handling the payments with the Payment Service Providers or, also common nowadays, receiving receipts from the mobile stores (Apple, Google, Amazon), validating those receipts, and acting accordingly. This means that often backend services work along with other backend services (server-to-server communication).

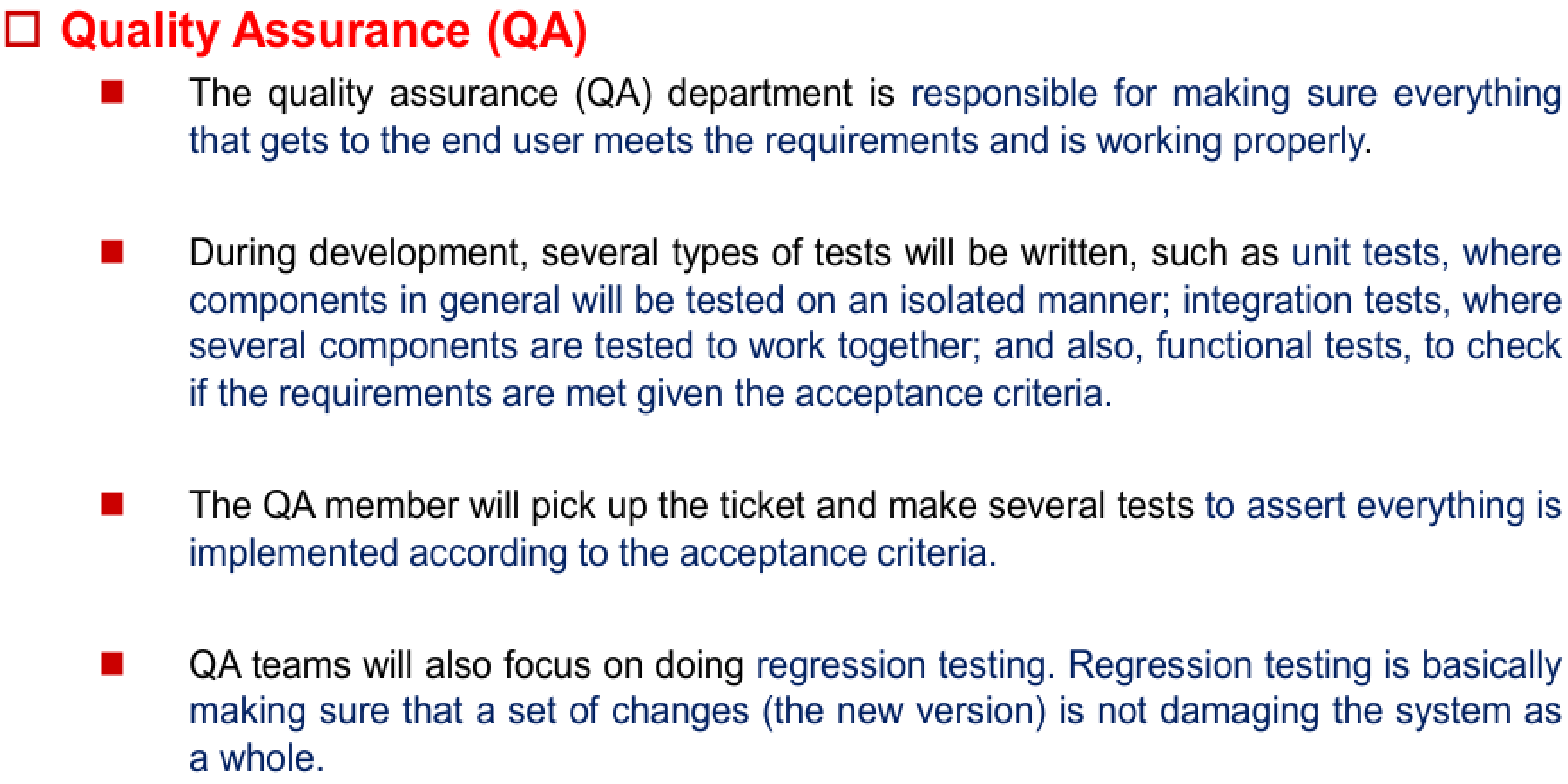
Roles and Responsibilities Cont.



□ Frontend

- The frontend application of a product is the one that is visible to the end-user.
- In general, when we refer to frontend we are thinking of a web application running on a browser. we'll be mentioning web applications running on browsers using JavaScript, HTML, and CSS.
- We can split the frontend work in two main modules: representation and logic. Representation is what the user sees, the interface, how the elements are rendered, and how to interact with them. The logic is everything else that makes it an application, such as fetching data, transforming them to present it to the user, and handling requests, states, validation of data input.
- There are even some cases where you will find people doing frontend and backend altogether; usually people with such abilities are called full-stack engineers or full-stack programmers, depending on the experience or academic degree.
- Usually this is the last development step for most of the user stories or features.

Roles and Responsibilities Cont.



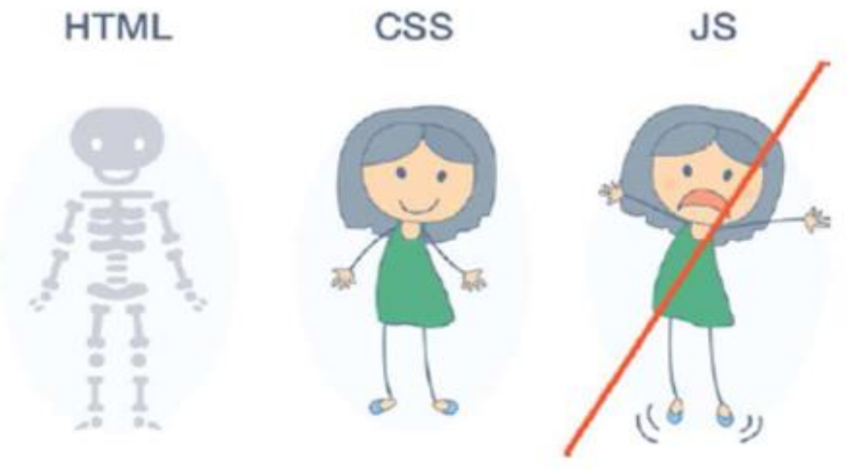
- **Quality Assurance (QA)**
  - The quality assurance (QA) department is responsible for making sure everything that gets to the end user meets the requirements and is working properly.
  - During development, several types of tests will be written, such as unit tests, where components in general will be tested on an isolated manner; integration tests, where several components are tested to work together; and also, functional tests, to check if the requirements are met given the acceptance criteria.
  - The QA member will pick up the ticket and make several tests to assert everything is implemented according to the acceptance criteria.
  - QA teams will also focus on doing regression testing. Regression testing is basically making sure that a set of changes (the new version) is not damaging the system as a whole.

Roles and Responsibilities Cont.

□ DevOps (development + operations).

- DevOps teams are responsible for all the operational aspects of the development and infrastructure. This means they are responsible for building the continuous integration and continuous delivery pipelines, managing the servers, performing migrations, and doing the actual deployments.
- This role is often mistaken with System Administrators. System administrators also deal with the all the entities that keep the company running—for example, the office network and internal systems used on a daily basis.

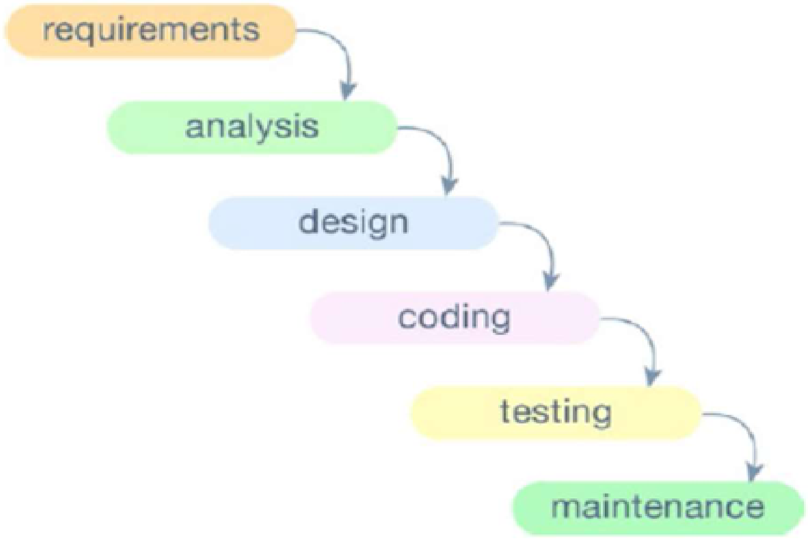
UI engineering does not include the dynamic part of the frontend Development



Methodologies: Waterfall

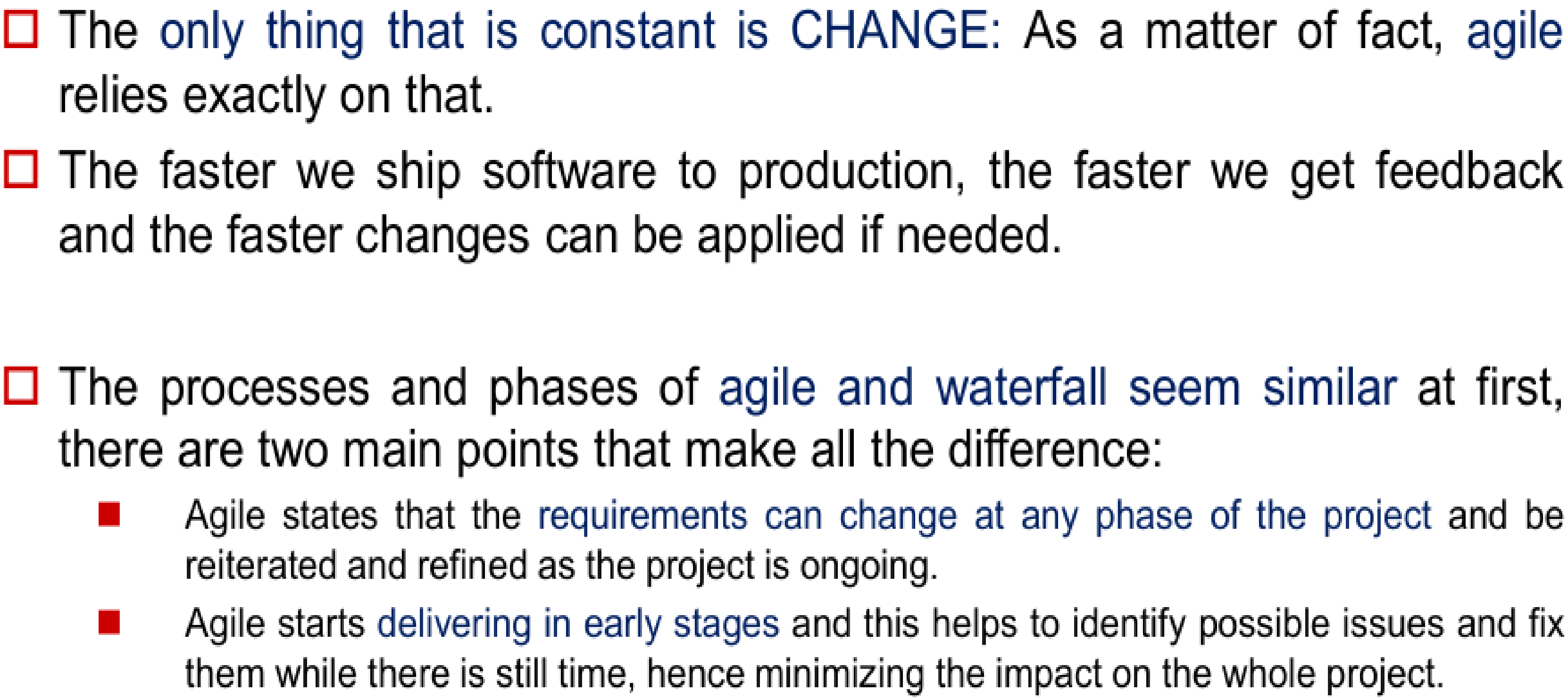
- Waterfall software development model, from requirements to Maintenance

- Their implementation is sequential, meaning the next stage cannot be started before the previous one has been closed, documented, and approved.



- *It has been proven very effective for small projects that are not likely to change within their time of implementation.*
- *The client or the owner of the project will never see a working version before the project reaches the implementation phase.*

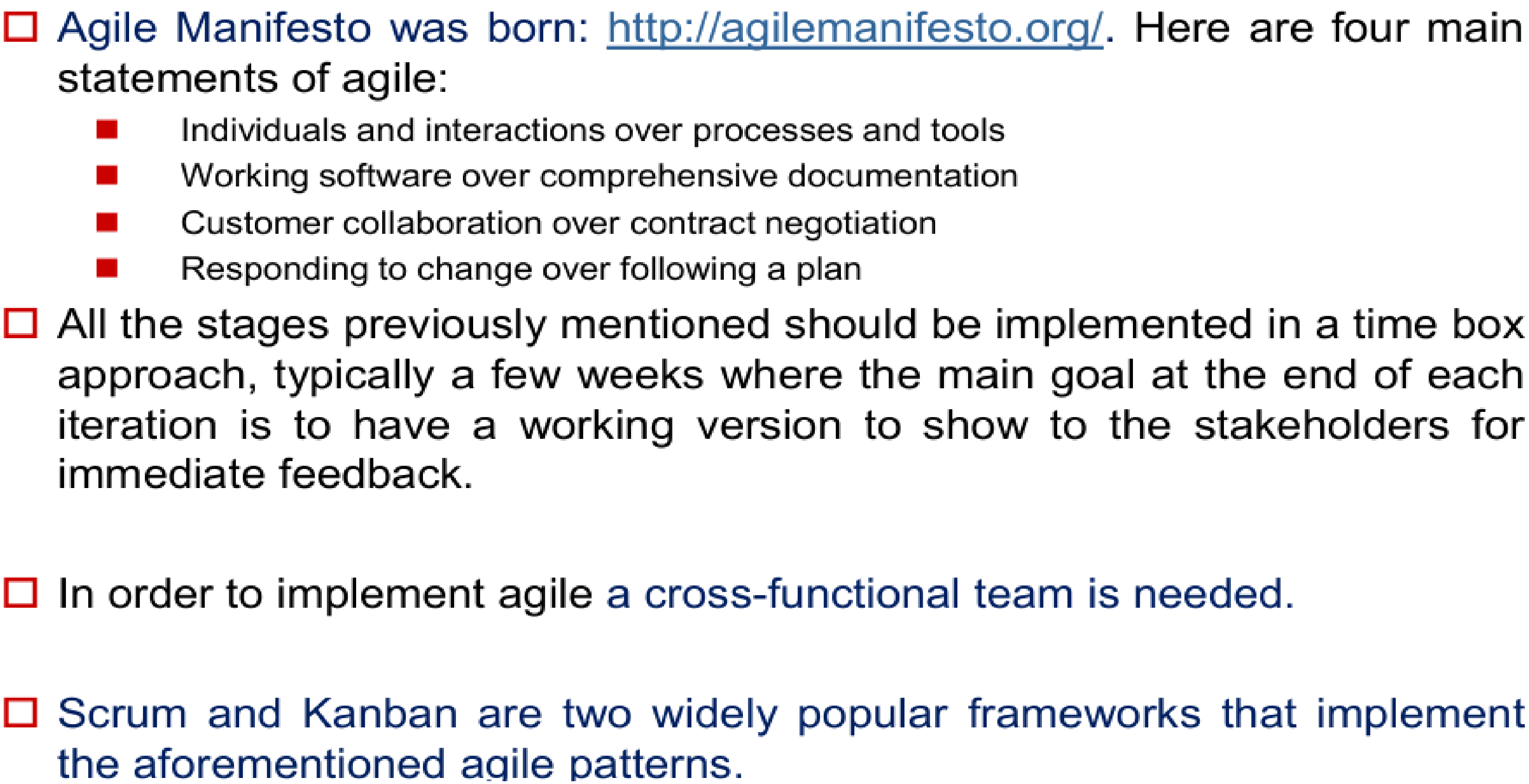
Methodologies: Agile



- The only thing that is constant is CHANGE: As a matter of fact, agile relies exactly on that.
- The faster we ship software to production, the faster we get feedback and the faster changes can be applied if needed.

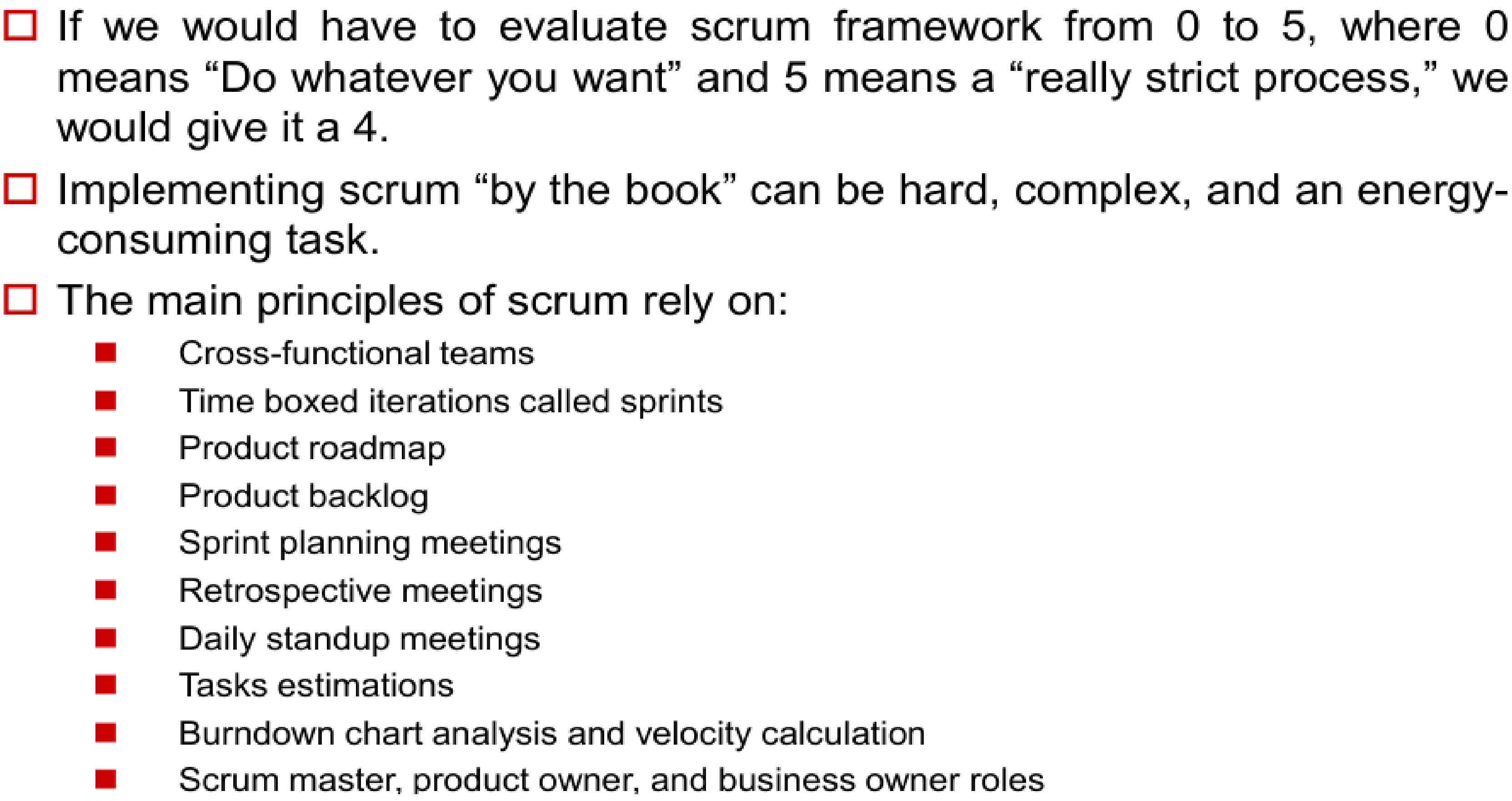
- The processes and phases of agile and waterfall seem similar at first, there are two main points that make all the difference:
  - Agile states that the requirements can change at any phase of the project and be reiterated and refined as the project is ongoing.
  - Agile starts delivering in early stages and this helps to identify possible issues and fix them while there is still time, hence minimizing the impact on the whole project.

Methodologies: Agile Cont.



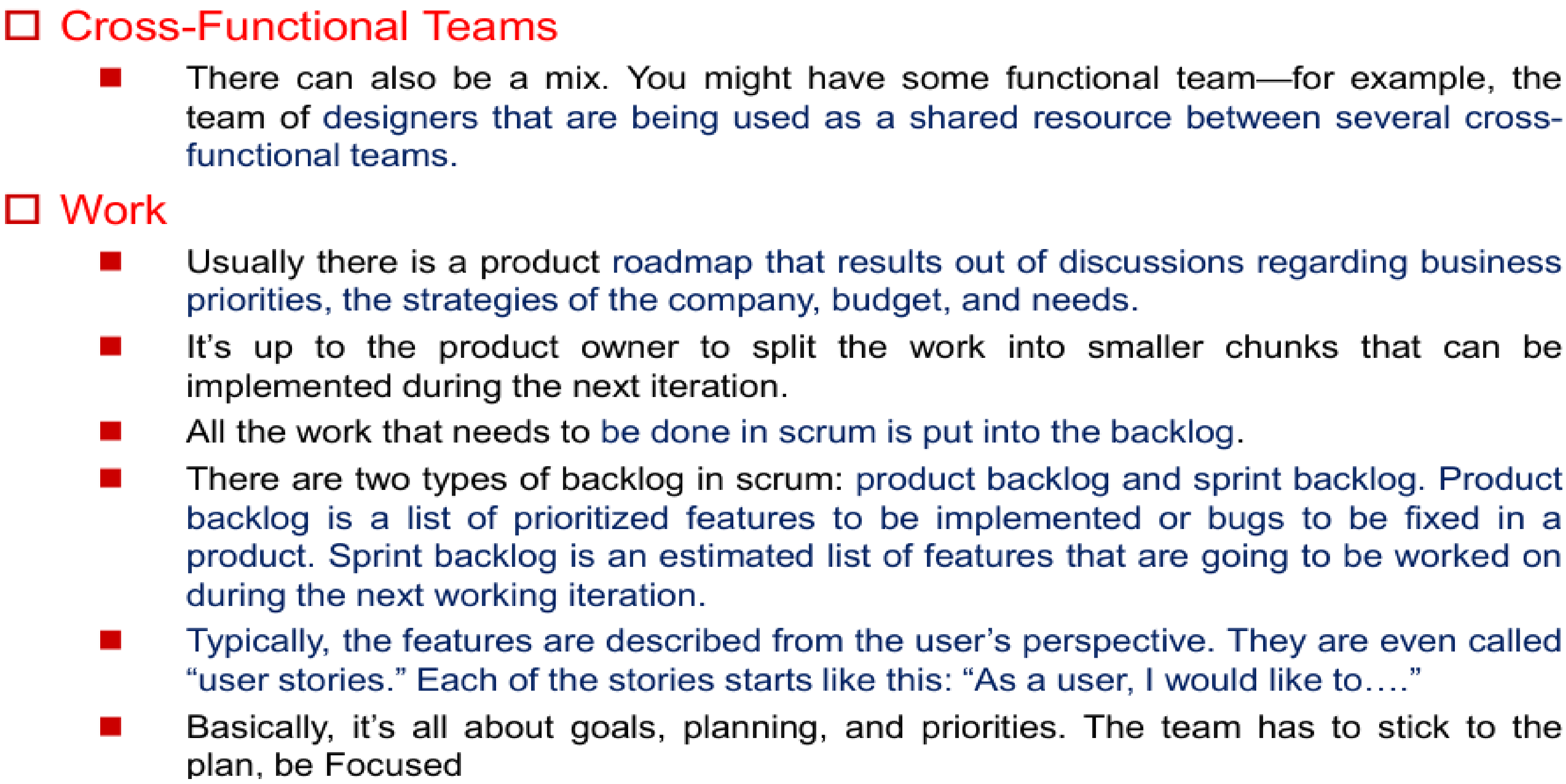
- Agile Manifesto was born: http://agilemanifesto.org/. Here are four main statements of agile:
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan
- All the stages previously mentioned should be implemented in a time box approach, typically a few weeks where the main goal at the end of each iteration is to have a working version to show to the stakeholders for immediate feedback.
- In order to implement agile a cross-functional team is needed.
- Scrum and Kanban are two widely popular frameworks that implement the aforementioned agile patterns.

Agile: Scrum



- If we would have to evaluate scrum framework from 0 to 5, where 0 means “Do whatever you want” and 5 means a “really strict process,” we would give it a 4.
- Implementing scrum “by the book” can be hard, complex, and an energy-consuming task.
- The main principles of scrum rely on:
  - Cross-functional teams
  - Time boxed iterations called sprints
  - Product roadmap
  - Product backlog
  - Sprint planning meetings
  - Retrospective meetings
  - Daily standup meetings
  - Tasks estimations
  - Burndown chart analysis and velocity calculation
  - Scrum master, product owner, and business owner roles

Agile: Scrum Cont.



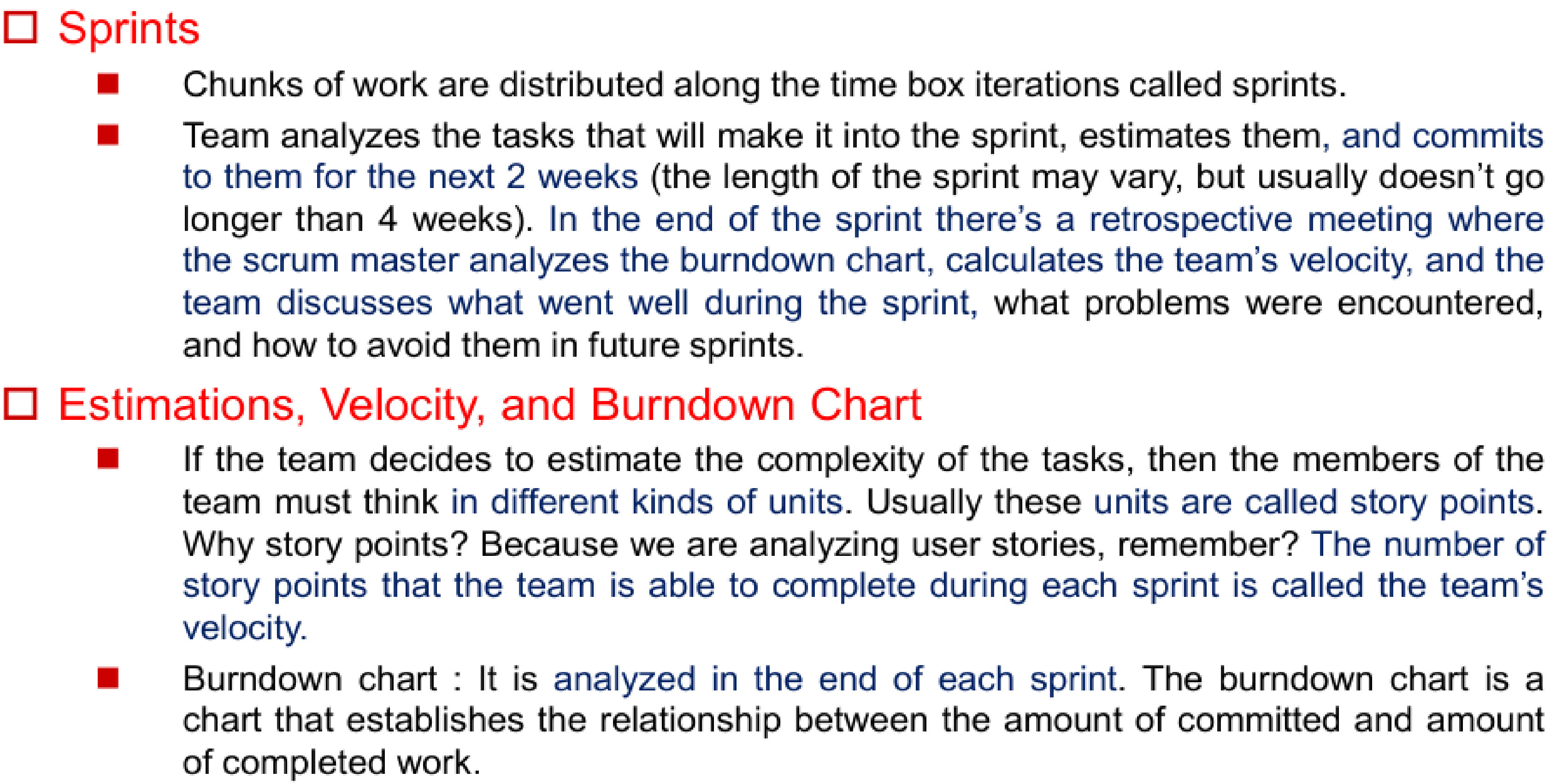
□ Cross-Functional Teams

- There can also be a mix. You might have some functional team—for example, the team of designers that are being used as a shared resource between several cross-functional teams.

□ Work

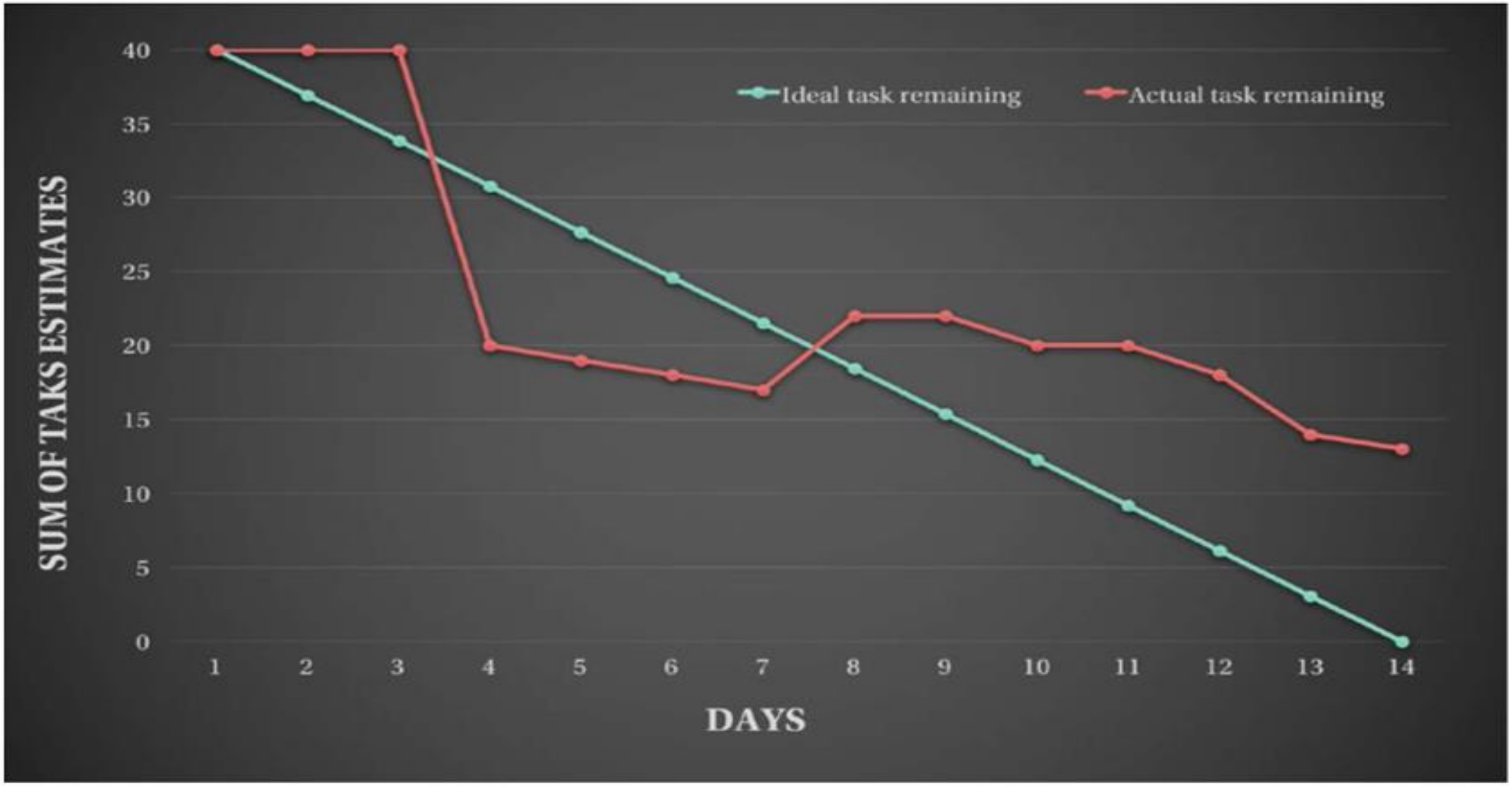
- Usually there is a product roadmap that results out of discussions regarding business priorities, the strategies of the company, budget, and needs.
- It's up to the product owner to split the work into smaller chunks that can be implemented during the next iteration.
- All the work that needs to be done in scrum is put into the backlog.
- There are two types of backlog in scrum: product backlog and sprint backlog. Product backlog is a list of prioritized features to be implemented or bugs to be fixed in a product. Sprint backlog is an estimated list of features that are going to be worked on during the next working iteration.
- Typically, the features are described from the user's perspective. They are even called "user stories." Each of the stories starts like this: "As a user, I would like to…."
- Basically, it's all about goals, planning, and priorities. The team has to stick to the plan, be Focused

Agile: Scrum Cont.

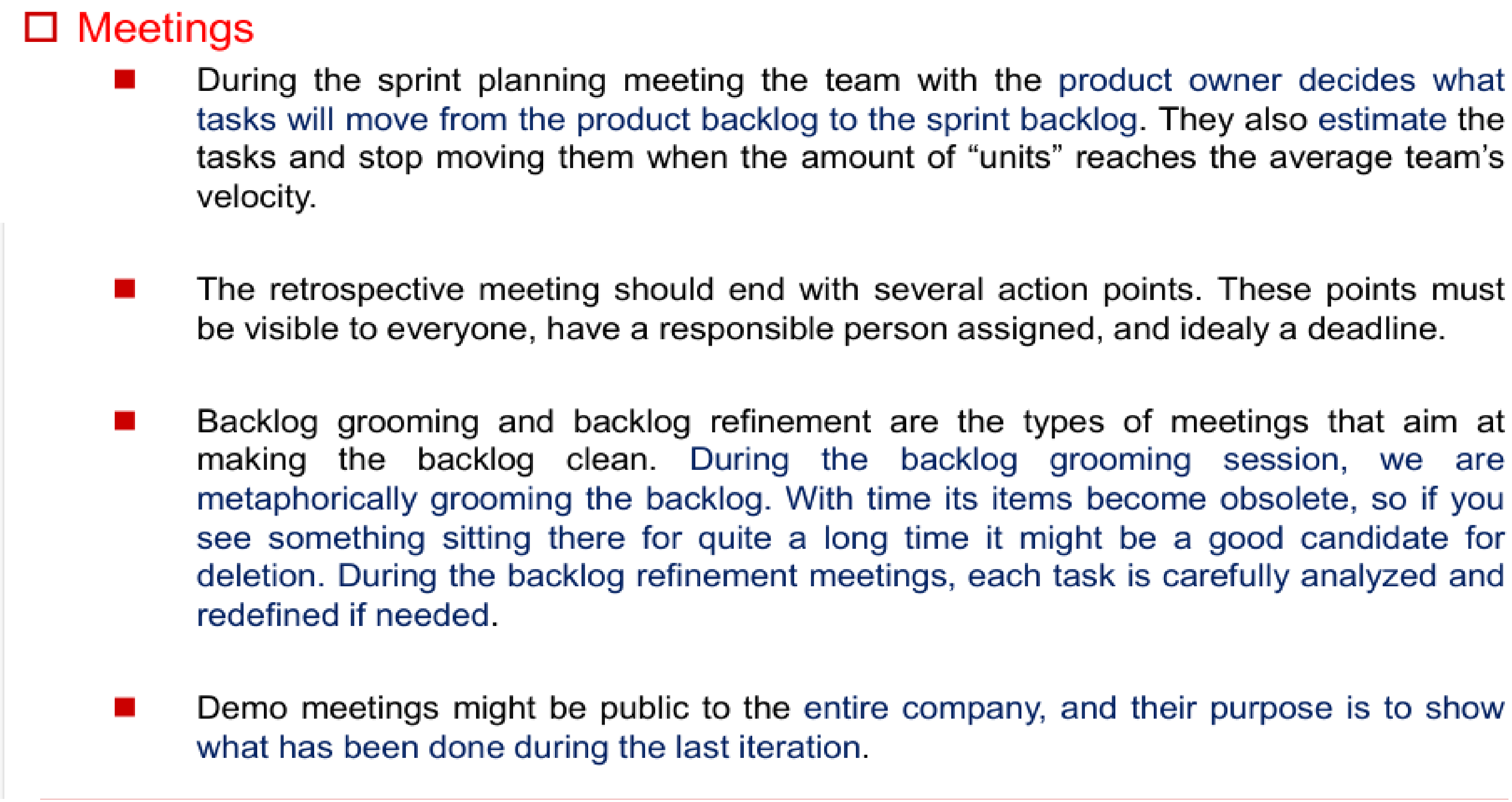


- □ Sprints
  - Chunks of work are distributed along the time box iterations called sprints.
  - Team analyzes the tasks that will make it into the sprint, estimates them, and commits to them for the next 2 weeks (the length of the sprint may vary, but usually doesn't go longer than 4 weeks). In the end of the sprint there's a retrospective meeting where the scrum master analyzes the burndown chart, calculates the team's velocity, and the team discusses what went well during the sprint, what problems were encountered, and how to avoid them in future sprints.
- □ Estimations, Velocity, and Burndown Chart
  - If the team decides to estimate the complexity of the tasks, then the members of the team must think in different kinds of units. Usually these units are called story points. Why story points? Because we are analyzing user stories, remember? The number of story points that the team is able to complete during each sprint is called the team's velocity.
  - Burndown chart : It is analyzed in the end of each sprint. The burndown chart is a chart that establishes the relationship between the amount of committed and amount of completed work.

Burndown chart



Agile: Scrum Cont.



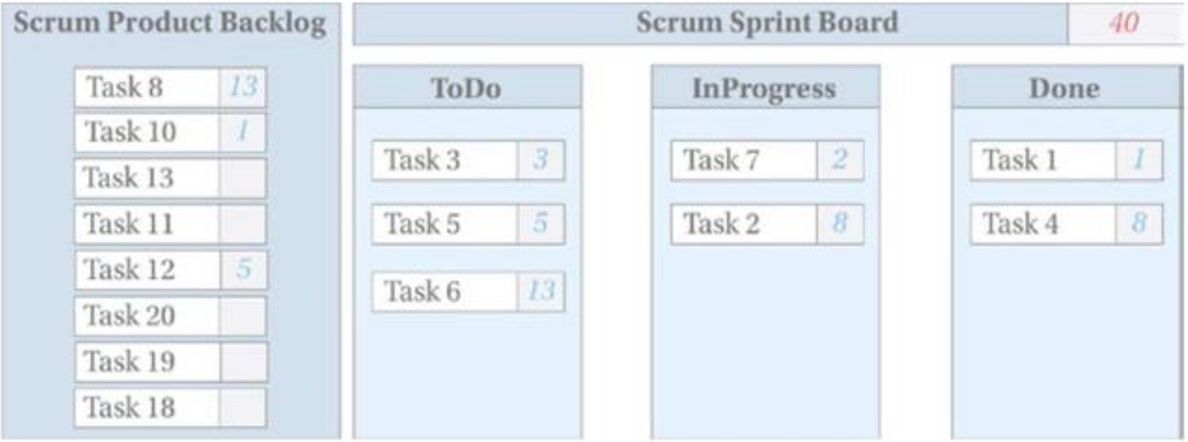
- Meetings
  - During the sprint planning meeting the team with the product owner decides what tasks will move from the product backlog to the sprint backlog. They also estimate the tasks and stop moving them when the amount of “units” reaches the average team’s velocity.
  - The retrospective meeting should end with several action points. These points must be visible to everyone, have a responsible person assigned, and idealy a deadline.
  - Backlog grooming and backlog refinement are the types of meetings that aim at making the backlog clean. During the backlog grooming session, we are metaphorically grooming the backlog. With time its items become obsolete, so if you see something sitting there for quite a long time it might be a good candidate for deletion. During the backlog refinement meetings, each task is carefully analyzed and redefined if needed.
  - Demo meetings might be public to the entire company, and their purpose is to show what has been done during the last iteration.

Agile: Scrum Cont.

□ Scrum Board

- Scrum board represents the state of both product backlog and sprint backlog. During the sprint the tasks can move from the "To Do" state to the "Done" state.

- Definition of done is a formal description of what criteria the task should meet to be considered done.



Scrum Product Backlog

Task 8 13
Task 10 1
Task 13
Task 11
Task 12 5
Task 20
Task 19
Task 18

Scrum Sprint Board 40

ToDo
Task 3 3
Task 5 5
Task 6 13

InProgress
Task 7 2
Task 2 8

Done
Task 1 1
Task 4 8

Scrum sprint board

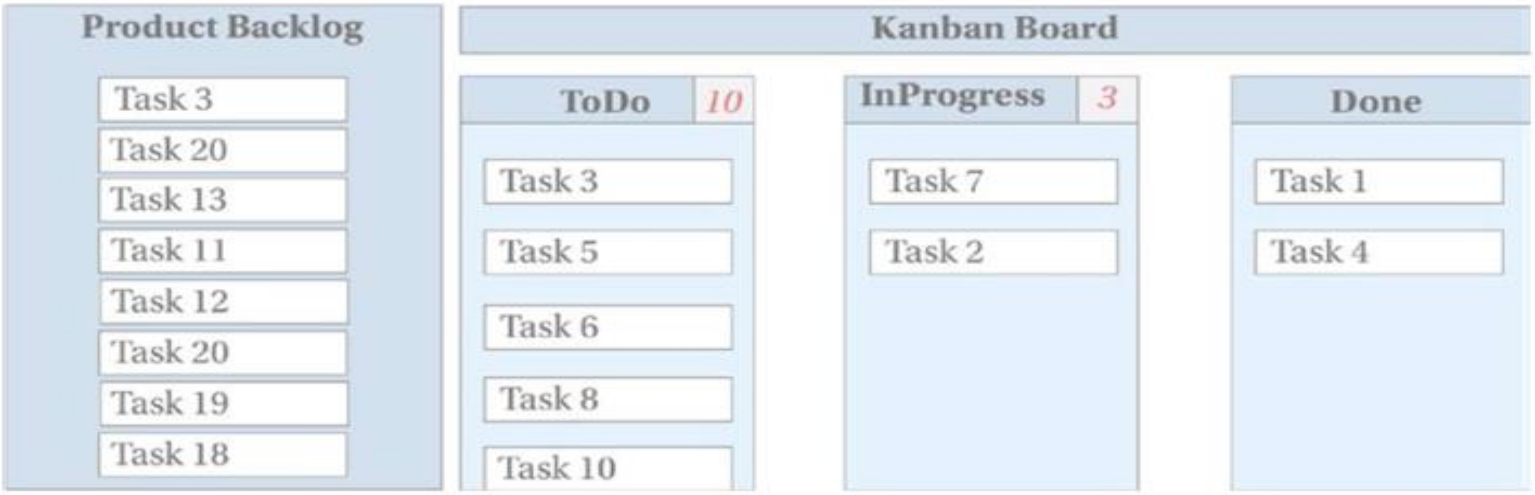
Agile: Kanban

- When we started discussing scrum, we evaluated it from 0 ("Do whatever you want") to 5 ("Really strict process") and gave it 4. Kanban, in our opinion, deserves a 2 on this scale.
- There are no must-have meetings, no special roles, and no time boxed iterations in Kanban. There are no estimations and no velocity calculations. The only restriction that you can have in Kanban is the limit of tasks per task stage.
- Planning
  - Even though Kanban doesn't have sprints, such as sprint planning meetings, Kanban teams still do planning. Remember, he who doesn't plan doesn't know how to manage time.
- Cycle Time
  - Remember the velocity metric from scrum? Kanban also has a very important metric called Cycle time. Cycle time is the time that takes a task to move from the "To Do" to "Done" column. Some studies also depict the metric called lead time. Lead time is the time between the creation of the task until it's actually done.

Agile: Kanban Cont.

□ Kanban Board

- Number of tasks per board, it limits them per column.



Product Backlog	Kanban Board		
	ToDo 10	InProgress 3	Done
Task 3	Task 3	Task 7	Task 1
Task 20	Task 5	Task 2	Task 4
Task 13	Task 6		
Task 11	Task 8		
Task 12	Task 10		
Task 20			
Task 19			
Task 18			

Kanban board

Thank You