

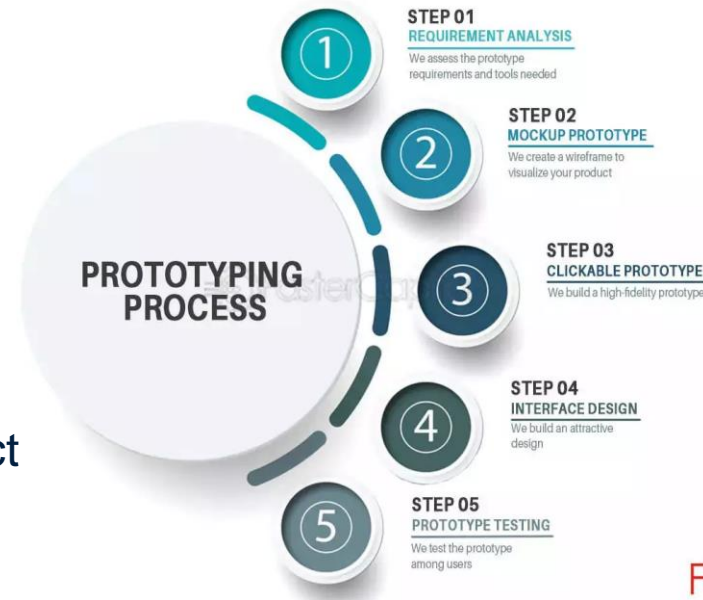
D0020E PROJECT IN COMPUTER SCIENCE 2022/2023 LECTURE 7.2: PROTOTYPING

Ulf Bodin
LTU 23.11.2022



Prototyping Software Products

- Prototyping is the development of a sample or prototype (in software engineering)
- Prototyping can be viewed as a step-by-step approach to the finished product
- A prototype becomes a finished product over the course of the project
- Prototyping allows for very early feedback from involved developers and especially from end-users



* Digital Leadership, Rapid Prototyping – Techniques, Benefits, Tools & Softwares. URL (accessed 2022-11-22): <https://digitalleadership.com/blog/prototyping/>

** Faster Capital, Software Prototype. URL (accessed 2022-11-22): <https://fastercapital.com/technical-making-a-prototype-for-your-product.html>

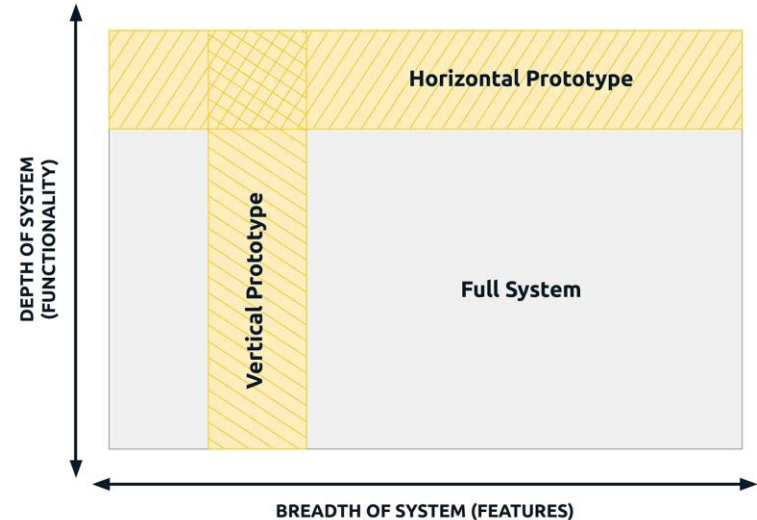
Types of Prototyping

1. Horizontal prototyping

- Focuses on a specific area of a software (e.g., GUI)
- Technical functionalities of the system and their implementation is missing
- GOAL: bring the user into contact with the product to obtain initial feedback

2. Vertical prototyping

- Picks a special area of a software & shows the interactions with other components
- GUI would already be mapped with data management and other parts of the system
- GOAL: explain complex functionalities and to let the user check the software in parts



➤ Digital Leadership

Prototyping approaches can also be separated based on application purposes

* Digital Leadership, Rapid Prototyping – Techniques, Benefits, Tools & Softwares. URL (accessed 2022-11-22): <https://digitalleadership.com/blog/prototyping/>

Types of Prototyping, cont.

3. Explorative prototyping

- Requirement profile of a software is clarified step by step by generating prototypes iteratively
- Functionalities are refined to assess whether the software solves the assumed problem
- Demonstrators are to make abstract requirements and problems clear and communicate them during development

4. Experimental prototyping

- A design is created with basic functions and tested regarding its feasibility
- The knowledge gained from this experiment or test is incorporated into the actual product
- Intended to help answer technical questions and check the project as such for its feasibility.

5. Evolutionary prototyping

- The software is created successively
- At each stage of development, feedback loops with users, developers and clients ensure that the final product meets the requirements profile.

* Digital Leadership, Rapid Prototyping – Techniques, Benefits, Tools & Softwares. URL (accessed 2022-11-22): <https://digitalleadership.com/blog/prototyping/>

Prototyping Tools

- GUI design
 - Mockingbird (<https://gomockingbird.com/home>)
 - Balsamiq (<https://balsamiq.com/>)
 - Pen and paper, Powerpoint
- Functionality
 - Powerpoint (example)
 - Wizard of Oz
 - Pen and Paper, Post-it notes, Storyboards
 - Video
 - Simple code

Balsamiq rapid prototyping

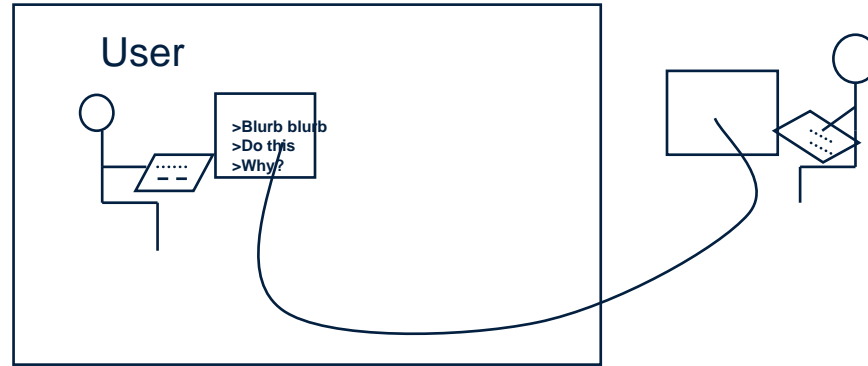


Powerpoint prototypes

- Add "functionality" by
 - Insert Hyperlink / Action
 - Place in document, Last slide viewed, First slide, etc.

Wizard of Oz

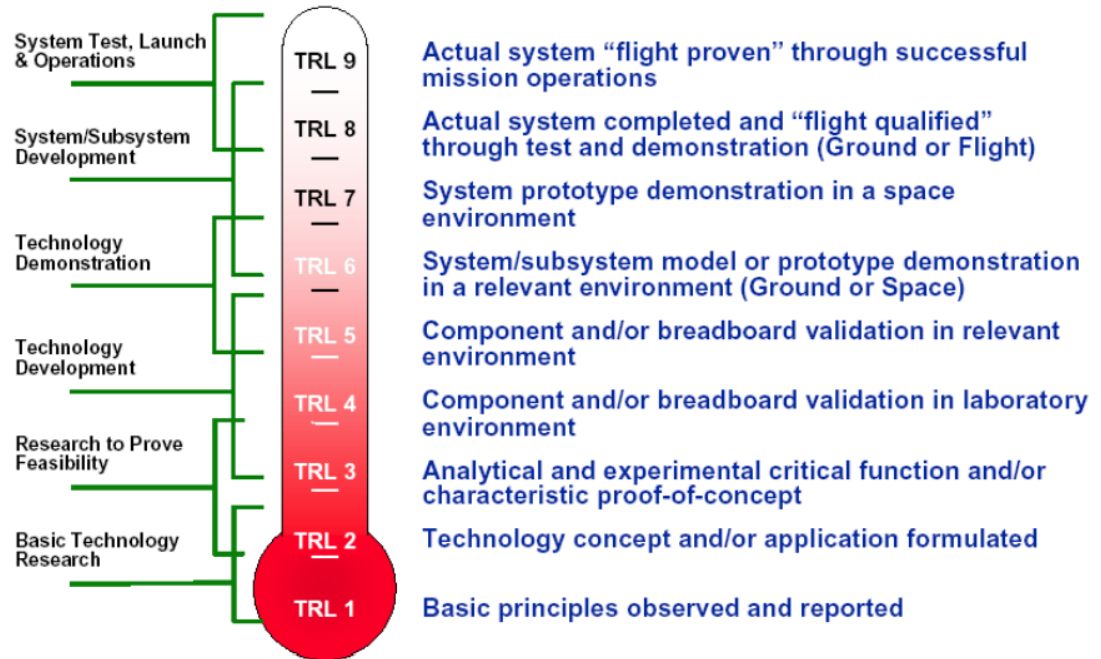
- Examples (<https://youtu.be/-SOeMA3DUEs>)



Technical Readiness Levels (TRLs)

NASA introduced the concept of TRLs as a means to track the progress of a new technology and determine how far it was along the path from just a novel idea to being ready for prime time

- NASA TRLs developed in the 1970th



* Armstrong JR. 6.4. 2 Applying Technical Readiness Levels to Software: New Thoughts and Examples. InINCOSE International Symposium 2010 Jul (Vol. 20, No. 1, pp. 838-845).

TRLs for software (NASA 2008)

1. **Scientific knowledge generated** underpinning basic properties of software architecture and mathematical formulation.
2. Practical application is identified but is speculative, no experimental proof or detailed analysis is available to support the conjecture. Basic properties of algorithms, representations and concepts defined. **Basic principles coded. Experiments performed with synthetic data.**
3. Development of limited functionality to **validate** critical properties and predictions using **non-integrated software components**.
4. **Key, functionally critical, software components are integrated, and functionally validated**, to establish interoperability and **begin architecture development**. Relevant environments defined and performance in this environment predicted.

* Armstrong JR. 6.4. 2 Applying Technical Readiness Levels to Software: New Thoughts and Examples. InINCOSE International Symposium 2010 Jul (Vol. 20, No. 1, pp. 838-845).

TRLs for software, cont.

5. **End-to-end software elements implemented and interfaced with existing systems/simulations conforming to target environment.** E2E software system, tested in relevant environment, meeting predicted performance. Operational environment performance predicted. Prototype implementations developed.
6. **Prototype implementations of the software demonstrated on full-scale realistic problems. Partially integrate with existing hardware/software systems.** Limited documentation available. Engineering feasibility fully demonstrated.
7. **Prototype software exists having all key functionality available for demonstration and test.** Well integrated with operational hardware/software systems demonstrating operational feasibility. **Most software bugs removed.** Limited documentation available.

* Armstrong JR. 6.4. 2 Applying Technical Readiness Levels to Software: New Thoughts and Examples. InINCOSE International Symposium 2010 Jul (Vol. 20, No. 1, pp. 838-845).

TRLs for software, cont.

8. All software has been thoroughly debugged and fully integrated with all operational hardware and software systems. All user documentation, training documentation, and maintenance documentation completed. **All functionality successfully demonstrated in simulated operational scenarios. Verification and Validation (V&V) completed.**
9. All software has been thoroughly debugged and fully integrated with all operational hardware/software systems. All documentation has been completed. **Sustaining software engineering support is in place. System has been successfully operated in the operational environment.**

* Armstrong JR. 6.4. 2 Applying Technical Readiness Levels to Software: New Thoughts and Examples. In INCOSE International Symposium 2010 Jul (Vol. 20, No. 1, pp. 838-845).

Minimum Viable Product (MVP)

- A “Minimum Viable Product” defines a certain development stage for a product or service. At this development stage, **it is possible for the first time to test the product/service under realistic conditions at the customer’s site and, above all, with the customer.**
- Only functions that are **absolutely necessary for the actual purpose** are implemented. This can also be just a single function, as long as it reflects the intended use case of the final product.
- Thus, an MVP is a version of a product with **just enough features** to be usable by early customers who can then provide feedback for future product development

* Digital Leadership, The Minimum Viable Product (MVP). URL (accessed 2022-11-22): <https://digitalleadership.com/blog/minimum-viable-product/>

Minimum Viable Product (MVP)

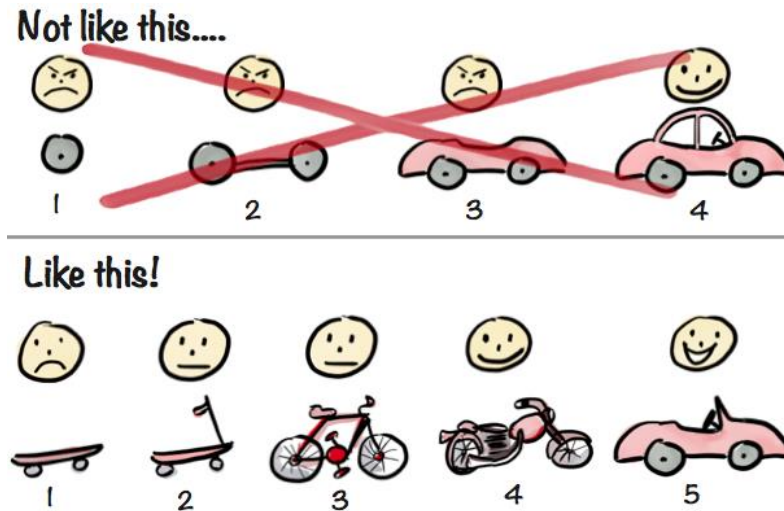


* CleverTap, What Is A Minimum Viable Product + Methodologies For Marketers. URL (accessed 2022-11-22): <https://clevertap.com/blog/minimum-viable-product/>

Minimum Viable Product (MVP)

Some alternative definitions

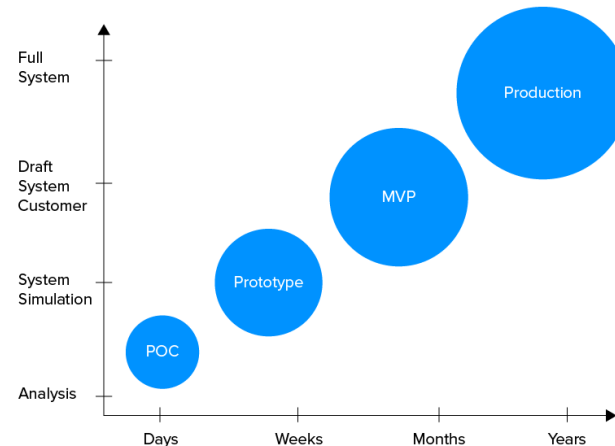
- The first version of a product
- A stripped-down version of a product
- A full-scale but simple product



Considerations	Minimum Viable Product	Proof of Concept	Prototype
What is the purpose of the development?	To find out where the product adds value and what its core features are	To decide if the idea will be feasible as a product	To examine how the product will work in greater depth.
Does the strategy provide the company with customers?	If the MVP is well thought out, customers can be attracted from launch	Proof of concepts are usually not offered up to customers to use	Prototypes are usually not launched for customers to use
What are some of the main benefits that come from the strategy?	Making money from day one and achieving customer retention. Understanding the market	Attracting seed-stage funding. Seeing if the idea will work technically.	Seeing how the product might look and feel. Visualizing how it might function
Is it used internally or externally?	The product is used externally with actual customers and feedback can be gained from them	Tends to only be used internally or sometimes with investors	Tends to only be used internally
Does it function for customers?	The MVP is a fully functioning working product which has the most critical features operating	No	No

Minimum Viable Product (MVP)

In any case, MVP is for the customers (e.g., to convince of usefulness) while PoC and prototypes tend to be for internal use (mainly) and feedback



* enkonix, What Is an MVP (And Why Is It Essential)?. URL (accessed 2022-11-22): <https://enkonix.com/blog/mvp/>

