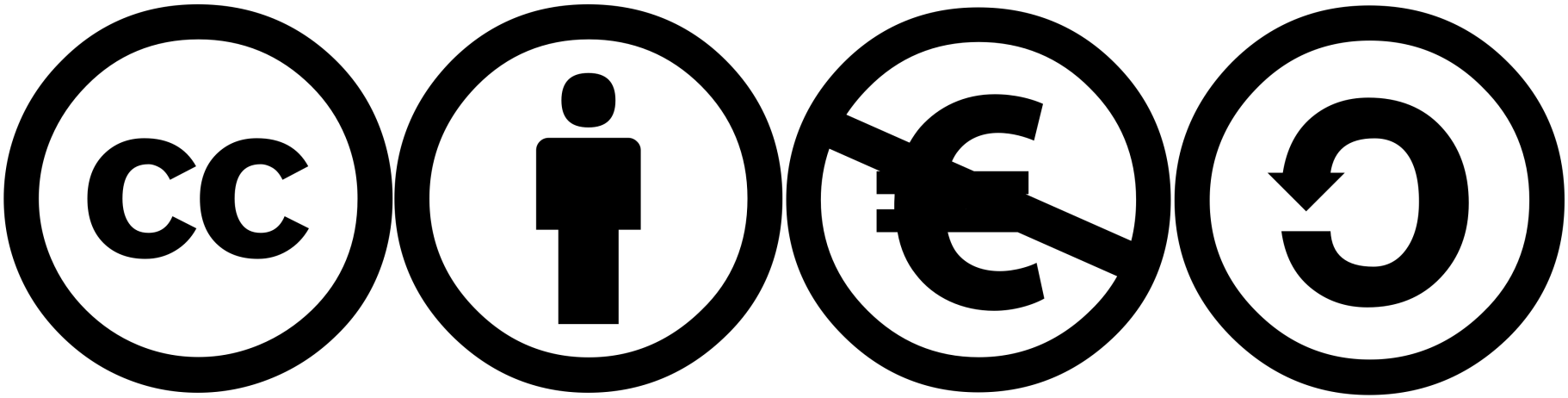
TB141Ic – ICT System Engineering and Rapid Prototyping

Jacopo De Stefani

Delft University of Technology, The Netherlands

13/02/2023

Licensing information



Teaching and support staff

Lecturer

Dr. Ir. Jacopo De Stefani
Lecturer at TUDelft, TBM-ESS-ICT

- E-mail: `J.deStefani@tudelft.nl`

Module manager

Prof.dr.ir. Marijn Janssen
Full professor at TUDelft, TBM-ESS-ICT

Course positioning in TBM Bachelor within I&C Track

Q1	Q2	Q3	Q4
TB111b Probleemanalyse Problem analysis	TB112c Systeemmodellering 1: modelleerprincipes Systems Modelling 1: Modelling Principles	TB133d Inleiding in programmeren in Python	TB113b Systeemmodellering 2: complexe systemen Systems Modelling 2: Complex Systems
TB131b Differentiaalvergelijkingen en Lineaire Algebra Differential equations and Linear Algebra	TB132b Multivariabele Analyse en Lineaire Algebra Multivariable Analysis and Linear Algebra	TB134a Statistiek en data-analyse Statistics and data analysis	TB135c Besliskunde Operational Research
TB121b Bestuur en recht I Governance and law I	TB122b Micro- en markteconomie Micro and market economy	TB141Ic ICT system engineering and rapid prototyping	TB142Ic Data science in Technische Bestuurskunde

I&C Track	
J1Q3	TB141Ic ICT system engineering and rapid prototyping
J1Q4	TB142Ic Data science in Technische Bestuurskunde
J2Q1	TB241IB Object-oriented programming
J2Q2	TB242Ia Interconnected Engineering Systems
J2Q4	TB243IB I & C: Risk and Control
J3Q3	TB341Ic I & C: Value Creation

Modeling / System Analysis
Mathematics
Economy
Specialization

The course structure

ICT System Engineering and Rapid Prototyping

The course structure

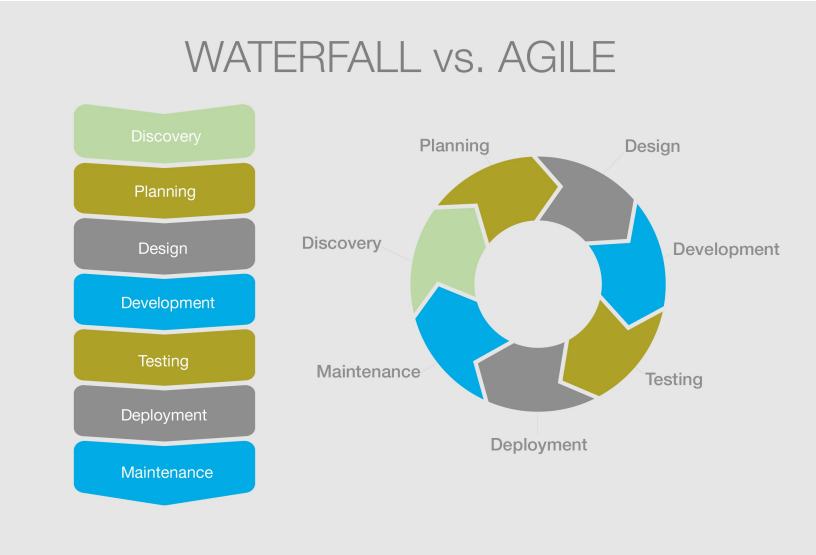
ICT System Engineering and Rapid Prototyping

Software development methodologies

The course structure

ICT System Engineering and Rapid Prototyping

Software development methodologies — Agile approaches and Mendix



Source: `https://www2.novacura.com/blog/agile-vs.-waterfall-for-erp-implementation`

- **Techniques**: Formal procedures for producing results using some well-defined notation
- **Methodologies:** Collection of techniques applied across software developments along following a given 'logic'
- **Tools:** Instruments or automated systems to accomplish a technique

The course structure

ICT System Engineering and Rapid Prototyping

Software development methodologies

Agile approaches and Mendix

- Low-code development platform
- Diagrams → Code/Data structures
- Sketches → UI

The course structure

$$\underbrace{\text{ICT System Engineering}}_{\text{Software development methodologies}} \text{ and } \underbrace{\text{Rapid Prototyping}}_{\text{Agile approaches and Mendix}}$$

In practice:

- 1^{st} half of the course: Methodology and basics of ICT (Theoretical Lectures + Formative Assignments)
- 2^{nd} half of the course: Rapid prototyping (Practical sessions + Summative Assignments)
- **Final Exam:** Week 10

Learning goals - ICT System Engineering

At the end of the course, the student should be able to:

- **LO1:** Describe the most common software design processes (at least Waterfall, Agile, Re-use based)
- **LO2:** Model real-life situations using UML diagrams (at least, use-cases, class and sequence diagrams)
- **LO3:** Explain the basic conceptual distinctions for characterizing IT systems, including the following:
  - **LO3.a:** Categorize programming languages in different levels of programming approaches (at least procedural, object-oriented, functional)
  - **LO3.b:** Define the basic components of computer hardware and hardware architectures
- **LO4:** Implement functional software using rapid prototyping approach and a low-code platform
- **LO5:** Identify the societal impact of the developed application.

Course calendar

Course Week	Class	Date	Topic	Class Type	Formative Assignment	Summative Assignment	Exam
1	1	13/02/2023	Course structure and Q/A	Theoretical			
	2	15/02/2023	Overview of Software Development Methodologies + Introduction to Waterfall	Theoretical	FA1		
2	3	20/02/2023	Requirements Engineering	Theoretical	FA1		
	4	23/02/2023	System Modeling - UML	Theoretical	FA1+FA2		
3	5	27/02/2023	Workshop: Modeling with UML	Theoretical	FA1+FA2		
	6	01/03/2023	Software and Hardware architectures	Theoretical	FA2+FA3		
4	7	06/03/2023	Programming Languages Overview	Theoretical	FA2+FA3		
	8	08/03/2023	Workshop: Agile Software Development / Rapid Prototyping in Practice + Introduction to Mendix + Project presentation + Mendix methodology	Theoretical	FA2+FA4		
5	9	13/03/2023			FA2+FA4		No Exam
	10	15/03/2023			FA2	SA1+SA2	
6	11	20/03/2023	MVC in Mendix: View - Pages	Practical		SA1+SA2	
	12	23/03/2023	MVC in Mendix: Model - Domain Model	Practical		SA1+SA2	
7	13	27/03/2023	MVC in Mendix: Controller I - Microflow	Practical		SA1+SA2	
	14	29/03/2023	MVC in Mendix: Controller II - Microflow	Practical		SA1+SA2	
8	15	28/03/2022	Basic security of the application in Mendix - Use cases / Guest Lecture	Practical		SA1+SA2+SA3	
	16	03/04/2023	Mendix integration with external platforms / Guest Lecture	Practical		SA1+SA2+SA3	
9	17	TBD	Project Discussion			SA3	
	?	TBD					
10	?	13/04/2023					
	?	19/04/2023	Final Exam				Final Exam

Proposed Calendar - 1st half - Methodologies

Course Week	Class	Date	Topic	Class Type	Formative Assignment	Summative Assignment	Exam
1	1	13/02/2023	**Course structure and Q/A**	Theoretical			
	2	15/02/2023	**Overview of Software Development Methodologies + Introduction to Waterfall**	Theoretical	**FA1**		
2	3	20/02/2023	**Requirements Engineering**	Theoretical	**FA1**		
	4	23/02/2023	**System Modeling - UML**	Theoretical	**FA1+FA2**		
	5	27/02/2023	**Workshop: Modeling with UML**	Theoretical	**FA1+FA2**		
3	6	01/03/2023	**Software and Hardware architectures**	Theoretical	**FA2+FA3**		
	7	06/03/2023	**Programming Languages Overview**	Theoretical	**FA2+FA3**		
4	8	08/03/2023	**Workshop: Agile Software Development / Rapid Prototyping in Practice + Introduction to Mendix + Project presentation + Mendix methodology**	Theoretical	**FA2+FA4**		

Formative assignments:

- **FA1:** Compare Waterfall and Agile software development approaches, Define and validate software engineering requirements
- **FA2:** Develop diverse Unified Modelling Language (UML) diagrams and able to use them for maintenance of programmable system
- **FA3:** Differentiate the different levels of programming (Object Oriented Programming (OOP), Functional Programming, Scripting, Low-Code)
- **FA4:** Distinguish the basic of computer hardware and architecture

Proposed Calendar - 2nd half - Rapid prototyping

Course Week	Class	Date	Topic	Class Type	Formative Assignment	Summative Assignment	Exam
5	**10**	15/03/2023			**FA2**	**SA1+SA2**	
6	**11**	20/03/2023	**MVC in Mendix: View - Pages**	Practical		**SA1+SA2**	
	12	23/03/2023	**MVC in Mendix: Model - Domain Model**	Practical		**SA1+SA2**	
7	**13**	27/03/2023	**MVC in Mendix: Controller I - Microflow**	Practical		**SA1+SA2**	
	14	29/03/2023	**MVC in Mendix: Controller II - Microflow**	Practical		**SA1+SA2**	
8	**15**	28/03/2022	**Basic security of the application in Mendix - Use cases / Guest Lecture**	Practical		**SA1+SA2+SA3**	
	16	03/04/2023	**Mendix integration with external platforms / Guest Lecture**	Practical		**SA1+SA2+SA3**	
9	**17**	TBD	**Project Discussion**			**SA3**	
	?	TBD					
10	?	13/04/2023					
	?	19/04/2023	**Final Exam**				**Final Exam**

Summative assignments:

- **SA1:** Allow students assemble functional software using rapid prototyping approach
- **SA2:** Deliver functional software using low-code platform
- **SA3:** Teach students how to address societal issues while blending programmable systems

Course schedule - In a nutshell

- **ICT System Engineering - Theoretical lectures (2 x week) - Week 1-4**
  - Monday: 10:45-12:30
  - Wednesday: 10:45-12:30
- **Rapid prototyping - Practical sessions (2 x week) - Week 5-10**
  - Monday: 10:45-12:30
  - Wednesday: 10:45-12:30
- **Office hours**
  - Send an e-mail to `J.deStefani@tudelft.nl` to schedule an appointment.

Important information

- Brightspace will be the main communication platform, hence keep an eye on your e-mail for announcments
- Relevant material for each lecture will be posted on Brightspace (and announced beforehand). You are advised to read the material before coming to the lecture.
- You are free to contact me anytime via e-mail/Teams. Keep in mind that, if you contact me outside working hours/working days, I will be reading (and treating your e-mail) during the following working day.

Tips & Tricks - Writing an email

From : `Jan.Jansen@tudelft.nl`

To : `J.deStefani@tudelft.nl`

Subject : [TB141] - Problem with Z

Dear Mr. De Stefani,

I am writing you concerning the topic X, you presented during class Y.
I did not understand the concept of Z...

–
Jan Janssen
TBM Student - 0012345

Tips & Tricks - Writing an email

From : `Jan.Jansen@tudelft.nl` ← *Use your official TUDelft account*

To : `J.deStefani@tudelft.nl`

Subject : [TB141] - Problem with Z

Dear Mr. De Stefani,

I am writing you concerning the topic X, you presented during class Y.
I did not understand the concept of Z...

–
Jan Janssen
TBM Student - 0012345

Tips & Tricks - Writing an email

From : `Jan.Jansen@tudelft.nl`

To : `J.deStefani@tudelft.nl`

Subject : [TB141] - Problem with Z ← *Use [TB141] in the subject and summarize your question/problem*

Dear Mr. De Stefani,

I am writing you concerning the topic X, you presented during class Y.
I did not understand the concept of Z...

–
Jan Janssen
TBM Student - 0012345

Tips & Tricks - Writing an email

From : `Jan.Jansen@tudelft.nl`

To : `J.deStefani@tudelft.nl`

Subject : [TB141] - Problem with Z

Dear Mr. De Stefani, ← *A little politeness goes a long way...*

I am writing you concerning the topic X, you presented during class Y.
I did not understand the concept of Z...

–
Jan Janssen
TBM Student - 0012345

Tips & Tricks - Writing an email

From : `Jan.Jansen@tudelft.nl`

To : `J.deStefani@tudelft.nl`

Subject : [TB141] - Problem with Z

Dear Mr. De Stefani,

I am writing you concerning the topic X, you presented during class Y.
I did not understand the concept of Z...
↑ *Explain your problem concisely, but give all the necessary references in the course.*

–
Jan Janssen
TBM Student - 0012345

Tips & Tricks - Writing an email

From : `Jan.Jansen@tudelft.nl`

To : `J.deStefani@tudelft.nl`

Subject : [TB141] - Problem with Z

Dear Mr. De Stefani,

I am writing you concerning the topic X, you presented during class Y.
I did not understand the concept of Z...

–
Jan Janssen ← *Don't forget the signature !*
TBM Student - 0012345

Assessment and key dates for the course

- **Summative Project: Rapid prototyping with Mendix** - 30% of final grade
  - **Group formation:** 14/03 → 17/03
  - **Mendix setup:** 14/03
  - **First deliverables:** 28/03 (Week 3.6)
  - **Final deliverables:** TBD (Week 3.8)
  - **Project presentation:** TBD (Week 3.9)

- **Written Exam** - 70% of final grade
  - **On campus/Online:** TBD (Week 3.10)

EvaSys - Quality of education

What would you change if you were the course manager?
What are strong points you would definitely keep?

Share it...

- With me, the lecturer
- With Curius, Simone van der Boon - `bachelor@curius.nl` - TPM Bsc Courses
- With your fellow students who join the CRG (student panel evaluation group, half way the period)
- Fill in the questionnaire at the end of this course (in your mailbox)

Any questions/comments?

Now, a little bit about you...


Break

What is an ICT system?

Information and Communication System

What is an ICT system?

Information and Communication **System**

System - Definition

A system is a group of interacting or interrelated elements that act according to a set of rules to form a unified whole. A system, surrounded and influenced by its environment, is described by its boundaries, structure and purpose and expressed in its functioning.

What is an ICT system?

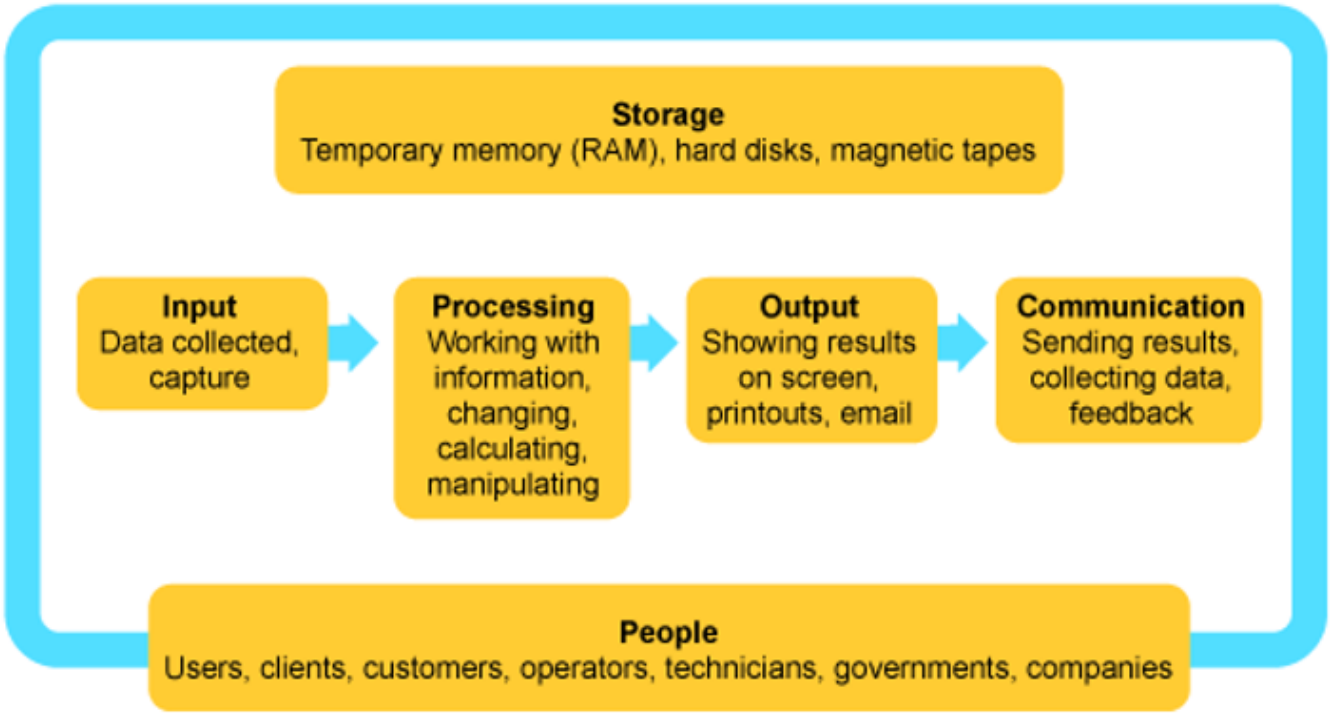
Information and Communication **System**

Working definition

A system of components directed by software that is capable of storing, retrieving, transmitting, communicating and manipulating information.

What is an ICT system?

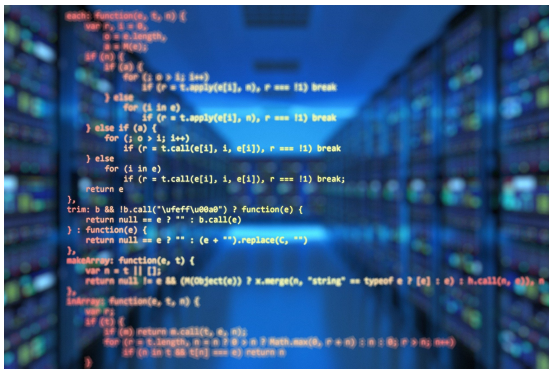
Information and Communication System



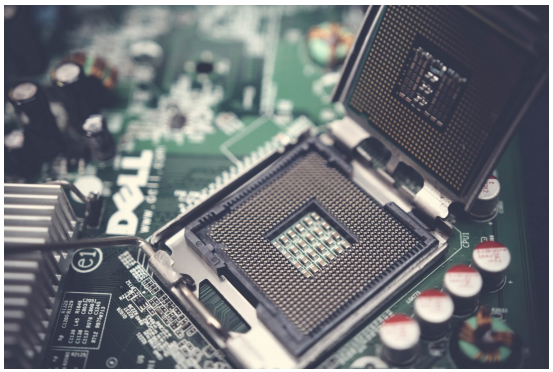
What is an ICT system?

Information and **Communication** System

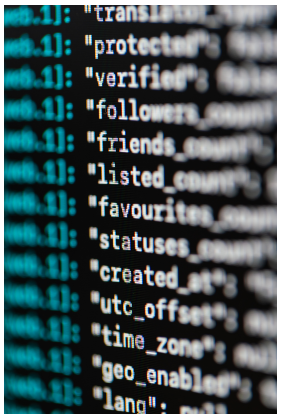
Software



Hardware



Data



What is an ICT system?

Information and Communication System

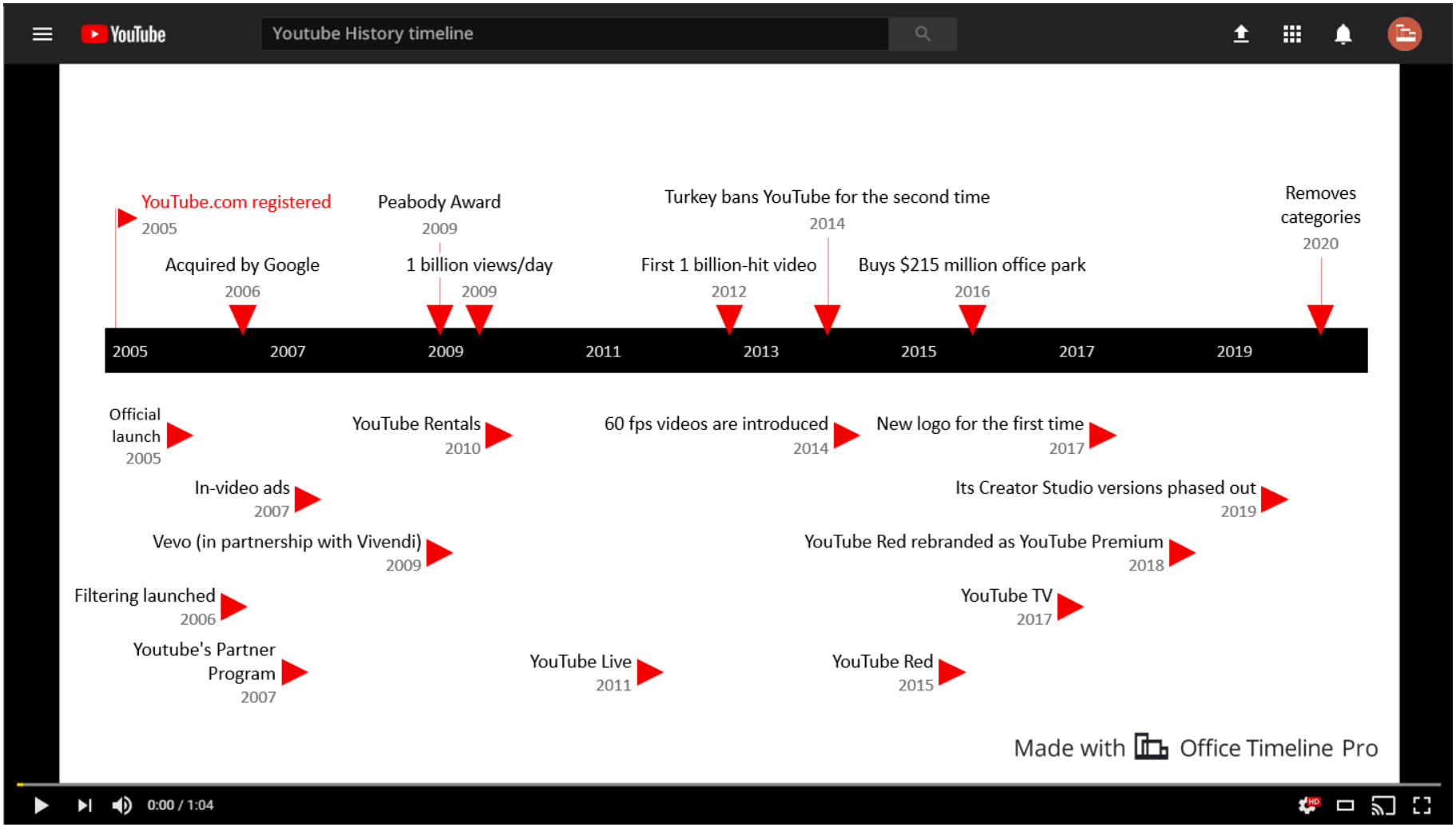
Telecommunications

People

What are examples of ICT systems, in your opinion?

Examples of ICT systems

Case study - YouTube (1)



Source: `https://www.officetimeline.com/blog/youtube-history-timeline`

Case study - YouTube (2)

- Worldwide adoption: Available in 103 countries
- Main revenue stream is driven by advertisement
- Recommender system proposes new videos in order increase permanence on the platform
- Led to the creation of new professions: content creators [1]
- Paradigm shift: From TV to streaming platforms [2]

[1] https://blog.youtube/inside-youtube/youtubes-creative-economy-small-businesses-big-impact/

[2] https://www.cnbc.com/2021/03/10/youtubes-mohan-to-pitch-advertisers-on-connected-tv-boom.html

Case study - YouTube (2)

- Worldwide adoption: Available in 103 countries
- Main revenue stream is driven by advertisement
- Recommender system proposes new videos in order increase permanence on the platform
- Led to the creation of new professions: content creators [1]
- Paradigm shift: From TV to streaming platforms [2]

In summary: socio-technical system (more than a simple IT system)

[1] https://blog.youtube/inside-youtube/youtubes-creative-economy-small-businesses-big-impact/

[2] https://www.cnbc.com/2021/03/10/youtubes-mohan-to-pitch-advertisers-on-connected-tv-boom.html

What are Information Systems?

Definition [3]

Information systems (IS) involve a variety of information technologies (IT) such as computers, software, databases, communication systems, the Internet, mobile devices and much more, to perform specific tasks, interact with and inform various actors in different organizational or social contexts.

[3] **7070407**

What are Information Systems?

Definition [3]

Information systems (IS) involve a variety of information technologies (IT) such as **computers, software, databases, communication systems, the Internet, mobile devices and much more**, to perform specific tasks, interact with and inform various actors in different organizational or social contexts.

[3] **7070407**

What are Information Systems?

Definition [3]

Information systems (IS) involve a variety of information technologies (IT) such as computers, software, databases, communication systems, the Internet, mobile devices and much more, **to perform specific tasks**, interact with and inform various actors in different organizational or social contexts.

[3] **7070407**

What are Information Systems?

Definition [3]

Information systems (IS) involve a variety of information technologies (IT) such as computers, software, databases, communication systems, the Internet, mobile devices and much more, to perform specific tasks, **interact with and inform various actors** in different organizational or social contexts.

[3] **7070407**

What are Information Systems?

Definition [3]

Information systems (IS) involve a variety of information technologies (IT) such as computers, software, databases, communication systems, the Internet, mobile devices and much more, to perform specific tasks, interact with and inform various actors **in different organizational or social contexts**.

[3] **7070407**

What are Information Systems?

Definition [3]

Information systems (IS) involve a variety of information technologies (IT) such as computers, software, databases, communication systems, the Internet, mobile devices and much more, to perform specific tasks, interact with and inform various actors in different organizational or social contexts.

Key elements

- Components (i.e. Hardware and Software)
- Tasks (i.e. functionalities)
- Actors (i.e. users)
- Context (i.e. environment)

[3] **7070407**

What is software?

- Sequences of instructions (algorithms, computer programs) that when executed provide desired features, function, and performance;
- Data structures that enable the programs to adequately manipulate information;
- Documentation that describes the operation and use of the programs.

```
1    # How to compute the sum of two numbers
2    a = input("Enter first number:")
3    b = input("Enter second number:")
4    print("The sum of a and b is:"+str(a+b))
```

Why is software important?

- Pervasive digitalization, both in the private and public sector. → Expenditure on software represents a significant fraction of GNP in all developed countries.
- Software can improve existing processes: reduce costs, increase speed, add reliability, repeatability, traceability, accurate, easier, productive, enjoyable, ...
- Or create new activities: see case study and gig/platform economy (Uber, Deliveroo? etc...)
- More and more systems are software controlled (transportation, medical, telecommunications, military, manifacturing, entertainment)
- In the current pandemic: shift from physical to digital activities.

What are the desirable properties of an IT/ICT system, in your opinion?

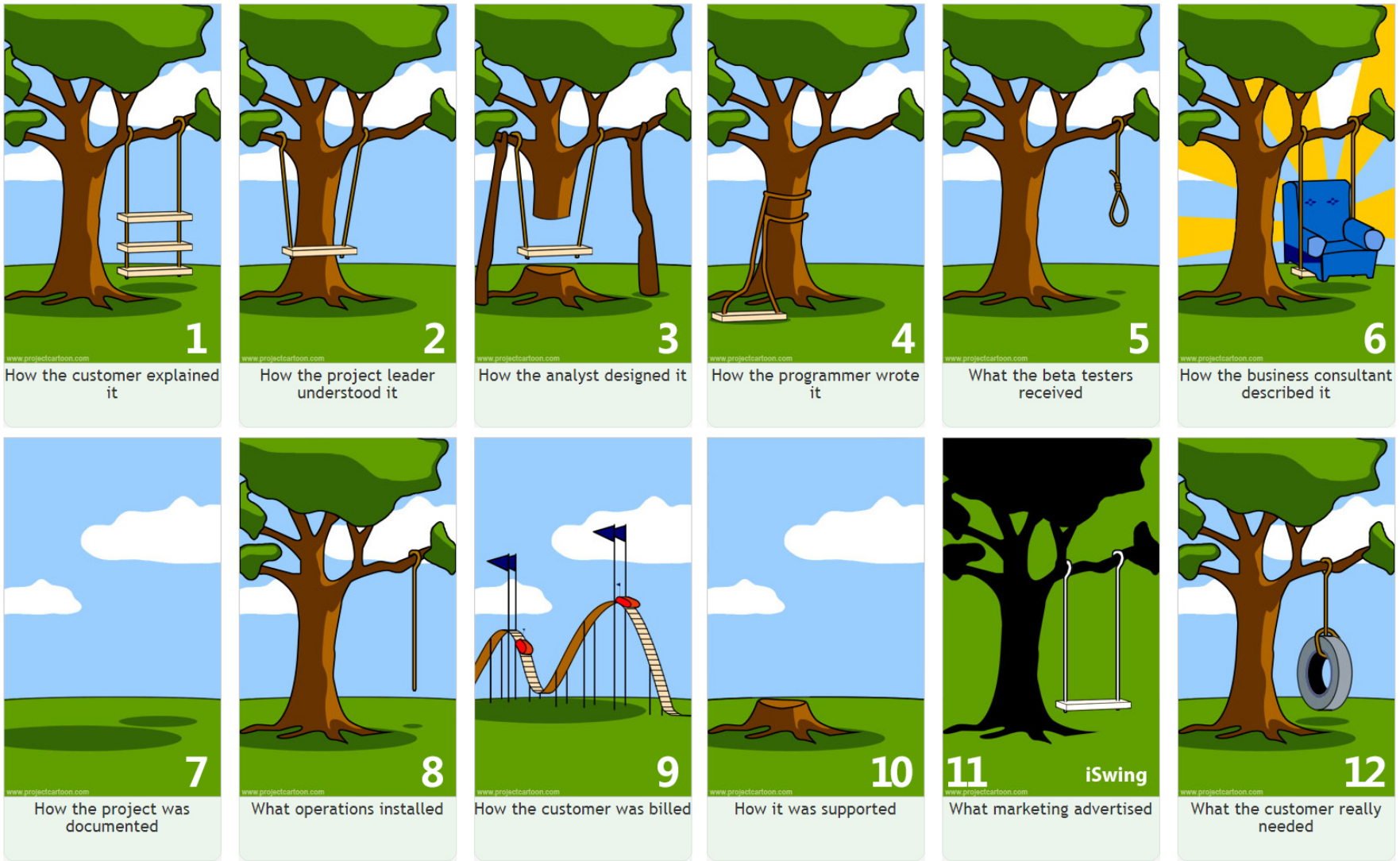
How do we achieve these desirable properties?

Software Engineering is a collection of techniques, methodologies and tools that help with the production of:

- a high quality software system
- with a given budget and specific constraints
- before a given deadline

while change occurs (in requirements, technologies and context)

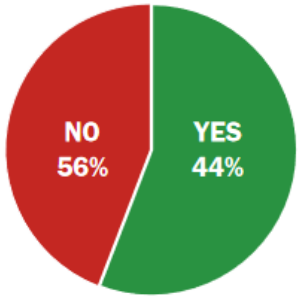
Why is engineering important?



Source: `https://www.zentao.pm/blog/tree-swing-project-management-tire-analogy-426.html`

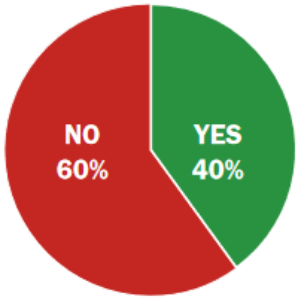
Why is engineering important?

ONBUDGET



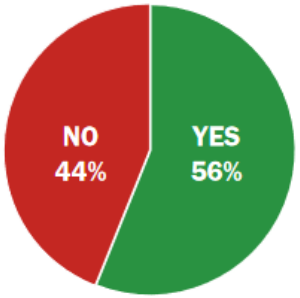
The percentage of projects that were OnBudget from FY2011–2015 within the new CHAOS database.

ONTIME



The percentage of projects that were OnTime from FY2011–2015 within the new CHAOS database.

ONTARGET



The percentage of projects that were OnTarget from FY2011–2015 within the new CHAOS database.

TRADITIONAL RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	39%	37%	41%	36%	36%
CHALLENGED	39%	46%	40%	47%	45%
FAILED	22%	17%	19%	17%	19%

The Traditional resolution of all software projects from FY2011–2015 within the new CHAOS database.

*All data, unless otherwise noted, represents results from FY2011-2015. The total number of software projects is 25,000-plus, with an average of 5,000 per yearly period.

Source: `https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf`

Why is engineering important?

- International research shows that for governmental ICT projects (world wide) with a budget higher than 10 million dollar the success rate (successful being defined as within budget and time schedule) is 7%. 36% of these projects fails completely (system never released).

  Source: `https://www.arnauddeklerk.nl/wp-content/uploads/2015/05/eindrapport-commissie-ict.pdf`

- Even when the project does fail to meet specifications, budgets are overrun for several different reasons.

  Source: `https://en.wikipedia.org/wiki/List_of_failed_and_overbudget_custom_software_projects`

Why do software engineering projects fail?

Why do software engineering projects fail?

- Unrealistic or unarticulated project goals
- Inaccurate estimates of needed resources
- Badly defined system requirements
- Poor reporting of the project's status
- Unmanaged risks
- Poor communication among customers, developers, and users
- Use of immature technology
- Inability to handle the project's complexity
- Sloppy development practices
- Poor project management
- Stakeholder politics
- Commercial pressures
- ...

How do we design a software-based system?

How do we design a software-based system? [4]

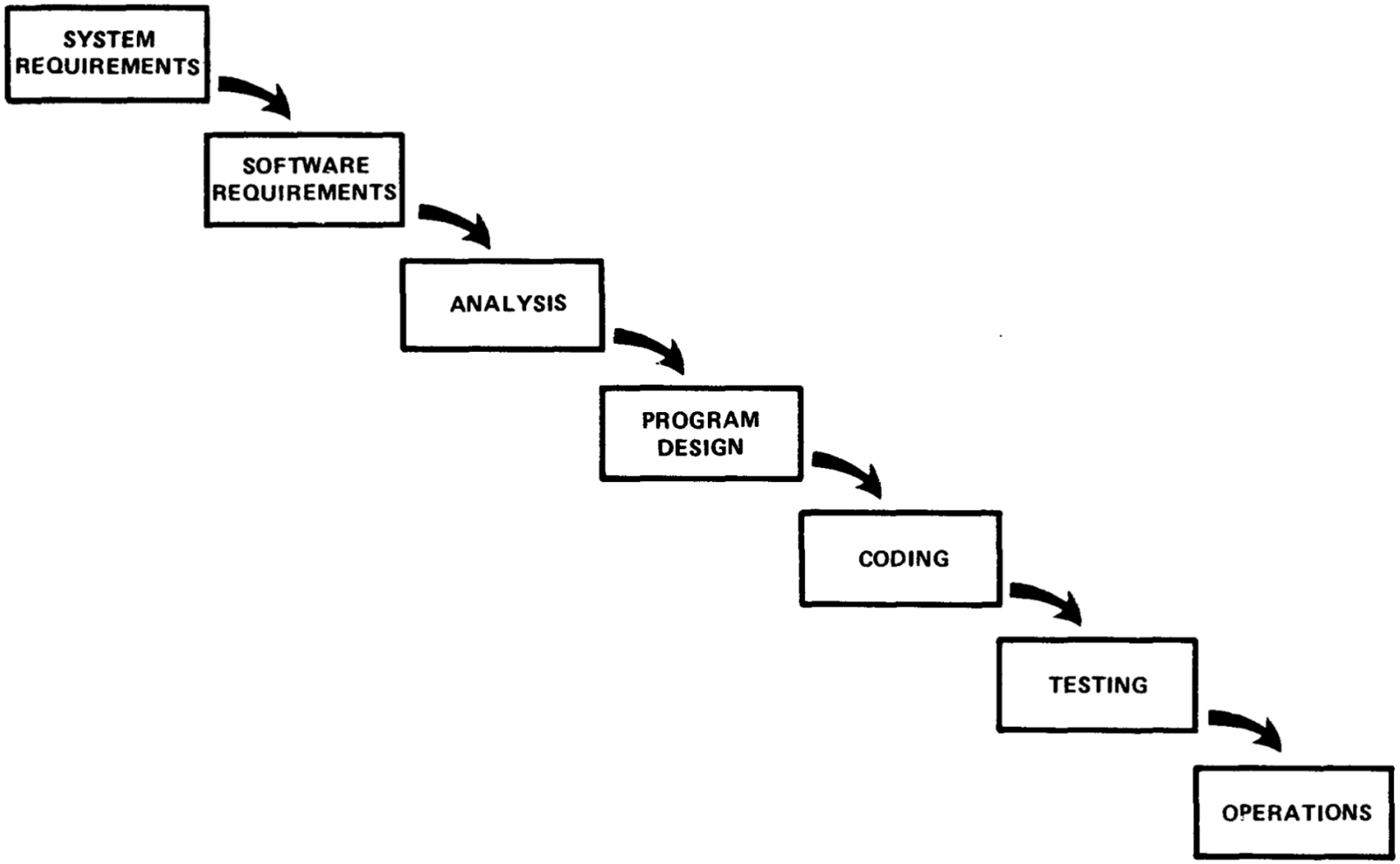


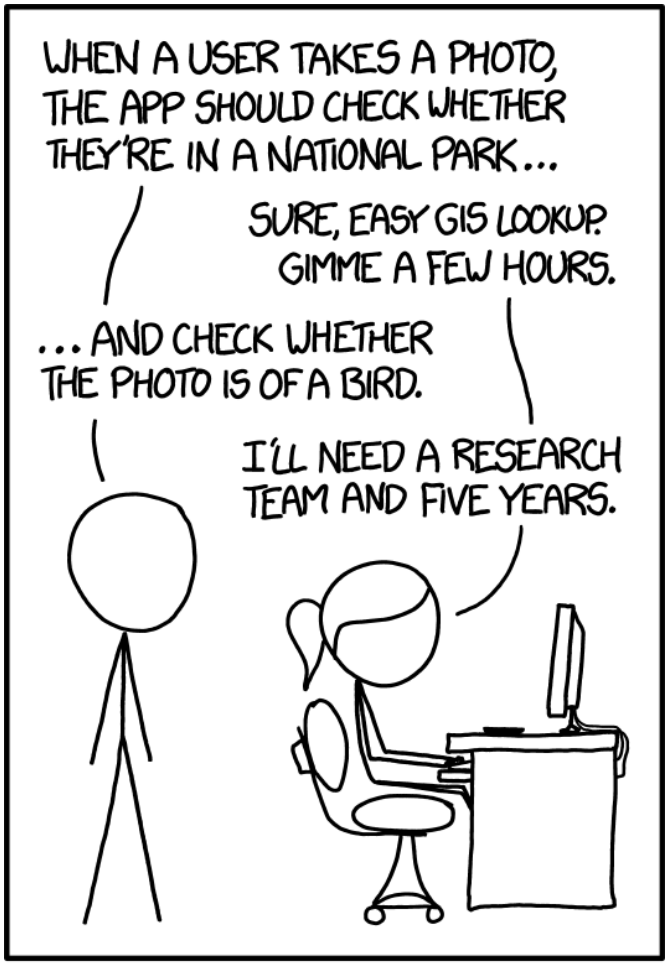
Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

[4] W. W. Royce (1987). "Managing the development of large software systems: concepts and techniques". In: *Proceedings of the 9th international conference on Software Engineering*, pp. 328–338

Up next:

- Overview of different software development methodologies
- Requirements engineering (system and software)
- Analysis and program design (via UML)
- Coding (using Mendix)

Thank you for your attention! Any questions/comments?



Source: `https://xkcd.com/1425/`

Appendix

FAQ about software and software engineering [5]

What is software?

Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.

What are the attributes of good software?

Good software should deliver the required functionality and performance to the user and should be maintainable, dependable, and usable.

[5] I. Sommerville (2011). "Software engineering 9th Edition". In: *ISBN-10* 137035152, p. 18

FAQ about software and software engineering [6]

What is software engineering?

Software engineering is an engineering discipline that is concerned with all aspects of software production.

What are the fundamental software engineering activities?

Software specification, software development, software validation, and software evolution.

What is the difference between software engineering and computer science?

Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.

[6] I. Sommerville (2011). "Software engineering 9th Edition". In: *ISBN-10* 137035152, p. 18

FAQ about software and software engineering [7]

What is the difference between software engineering and system engineering?

System engineering is concerned with all aspects of computer-based systems development including hardware, software, and process engineering. Software engineering is part of this more general process.

What are the key challenges facing software engineering?

Coping with increasing diversity, demands for reduced delivery times, and developing trustworthy software.

What are the costs of software engineering?

Roughly 60% of software costs are development costs; 40% are testing costs. For custom software, evolution costs often exceed development costs.

[7] I. Sommerville (2011). "Software engineering 9th Edition". In: *ISBN-10* 137035152, p. 18

FAQ about software [8]

What are the best software engineering techniques and methods?

While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another.

[8] I. Sommerville (2011). "Software engineering 9th Edition". In: *ISBN-10* 137035152, p. 18