EPFL

Software Requirements

Prof. George Candea
School of Computer & Communication Sciences

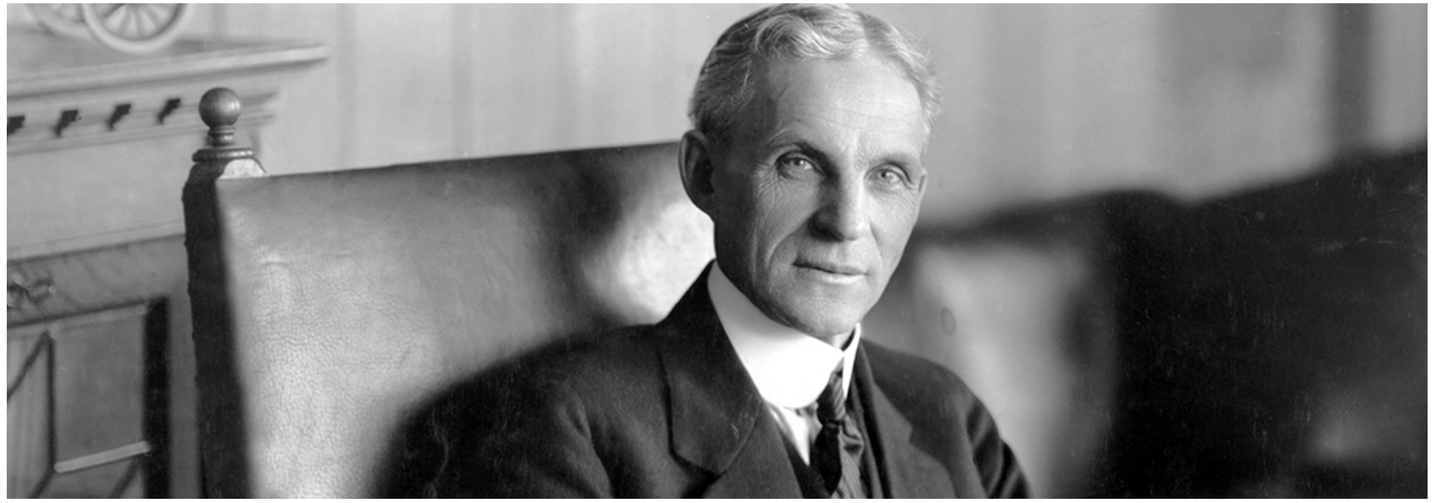
What is a requirement ?

- User problem = set of requirements
  - *features, performance criteria, security constraints, etc.*
- Requirements are not always objective

https://www.android.com/one/

What is a requirement ?

- User problem = set of requirements
  - *features, performance criteria, security constraints, etc.*
- Requirements are not always objective
- What users want ≠ what they need
- Hard to ask for something you've never seen



https://www.ecklers.com/history/ford_model_t

What is a requirement ?

- User problem = set of requirements
  - *features, performance criteria, security constraints, etc.*
- Requirements are not always objective
- What users want ≠ what they need
- Hard to ask for something you've never seen
- True needs vs. luxury needs

How do we define requirements ?

- Talk to users
- Requirements elicitation
- Persona = fictional user representing a group

Alice is a manager who uses her phone all the time between meetings, mainly for emails and calendar, and who also uses applications developed by her company such as for reporting her time.

Bob is an office worker who rarely uses his phone for work, but frequently browses the Web and listens to podcasts on his commute to and from work.

User stories

- User story = informal description in natural language
  - *through the eyes of a future user*
- Does not dictate implementation

As a [role], I want to [action], so that [reason]

As an administrator,
I want to authenticate to the system,
so that I can use the admin interface

Structured definitions

Given [context], when [smthg happens], then [reaction]

```
Feature: Learn about requirements

Scenario: Read notes
  Given the prof posted notes and student has not read them yet
  When the student has free time
  Then the student reads the notes

Scenario: Do exercises
  Given the student has read the notes
  When the student has free time
  Then the student does the exercises
```

Validating and prioritizing requirements

- Validation = do requirements correspond to user needs ?
- Can validate "in the wild"
- Prioritization
  - P0 (highest) / P1 (medium) / P2 (lowest)
  - must-have / should-have / could-have / won't-have (MoSCoW)
- The list of features is dynamic

P0s get done, P1s might get done, P2s never get done

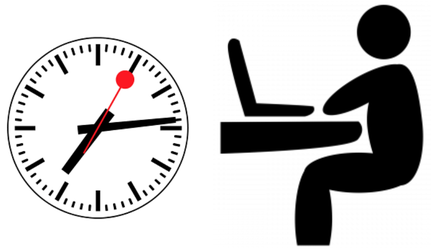
could-have turns into won't-have after a few months

Writing software that satisfies requirements

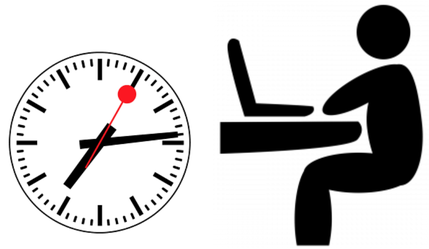
- User story ⟶ one or more tasks

As an administrator,
I want to authenticate to the system,
so that I can use the admin interface

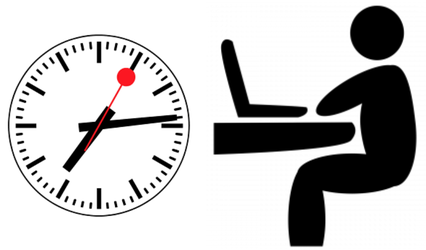
Choose auth service



Integrate login with service



Retrieve account settings



...