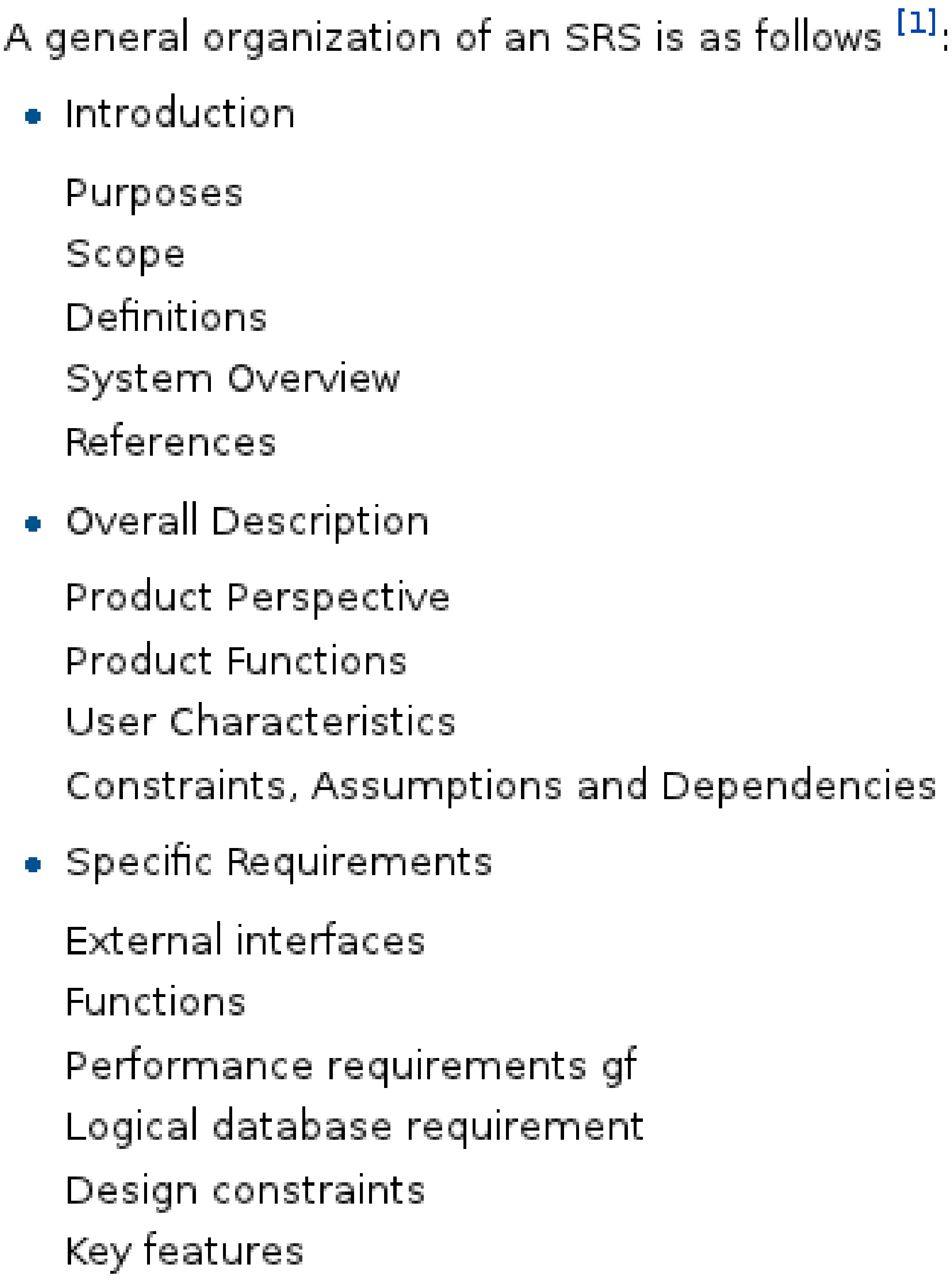
Task 4 : Project SRS Elicitation

- SE uses declarative approach(es) to define
  - what we want
  - what the constraints are
- Software Requirements Specification (SRS)
  - requirements
    - what we want / form of solution
    - *what* it should do (and not)
  - specifications
    - constraints, simplifications, etc.
    - what the constraints are / *how* it should do it (and not), but very light on implementation-level details

data

control

Task 4 : Project SRS Elicitation



A general organization of an SRS is as follows [1]:

- Introduction

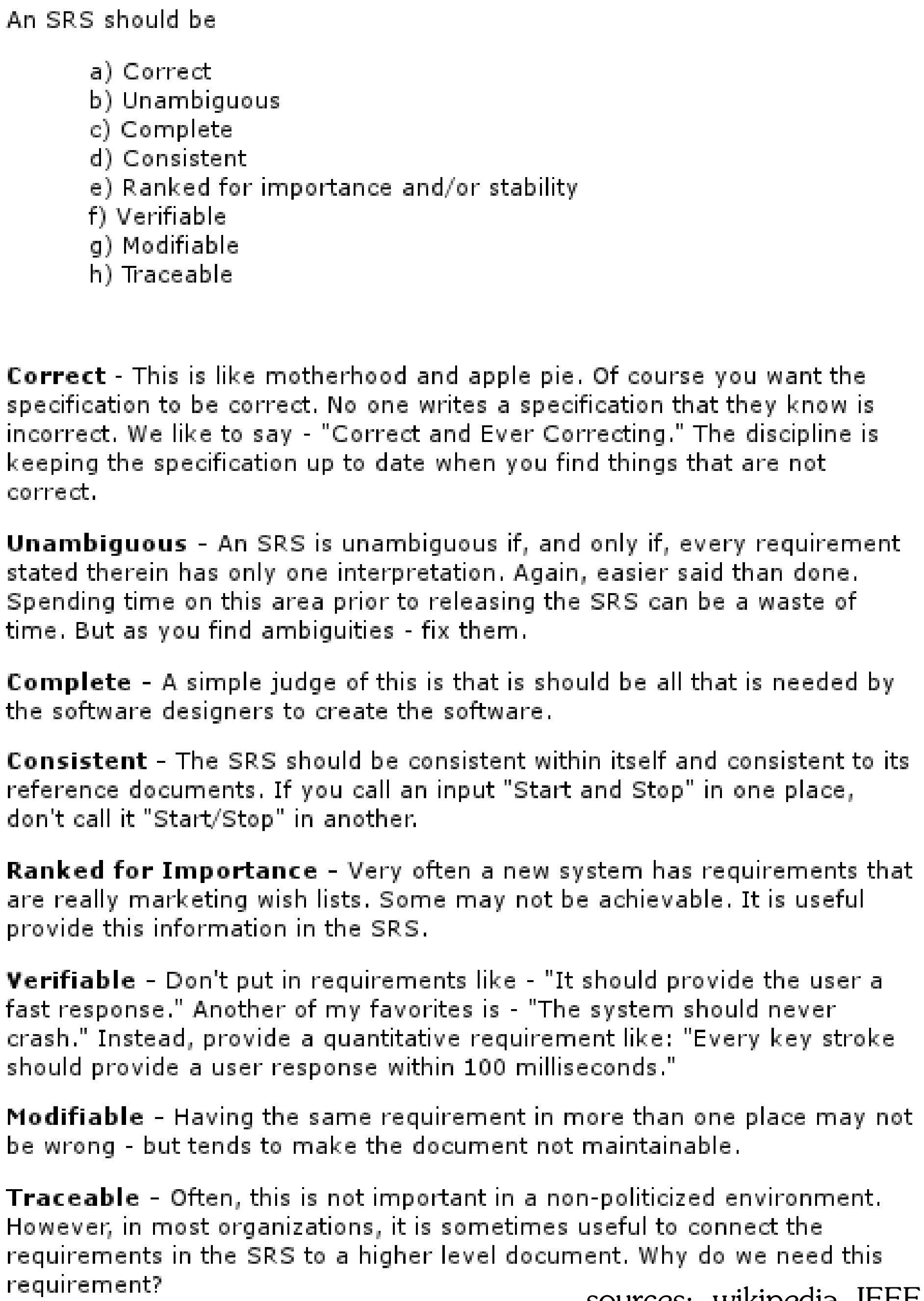
  Purposes
  Scope
  Definitions
  System Overview
  References

- Overall Description

  Product Perspective
  Product Functions
  User Characteristics
  Constraints, Assumptions and Dependencies

- Specific Requirements

  External interfaces
  Functions
  Performance requirements gf
  Logical database requirement
  Design constraints
  Key features



An SRS should be

a) Correct
b) Unambiguous
c) Complete
d) Consistent
e) Ranked for importance and/or stability
f) Verifiable
g) Modifiable
h) Traceable

Correct - This is like motherhood and apple pie. Of course you want the specification to be correct. No one writes a specification that they know is incorrect. We like to say - "Correct and Ever Correcting." The discipline is keeping the specification up to date when you find things that are not correct.

Unambiguous - An SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation. Again, easier said than done. Spending time on this area prior to releasing the SRS can be a waste of time. But as you find ambiguities - fix them.

Complete - A simple judge of this is that is should be all that is needed by the software designers to create the software.

Consistent - The SRS should be consistent within itself and consistent to its reference documents. If you call an input "Start and Stop" in one place, don't call it "Start/Stop" in another.

Ranked for Importance - Very often a new system has requirements that are really marketing wish lists. Some may not be achievable. It is useful provide this information in the SRS.

Verifiable - Don't put in requirements like - "It should provide the user a fast response." Another of my favorites is - "The system should never crash." Instead, provide a quantitative requirement like: "Every key stroke should provide a user response within 100 milliseconds."

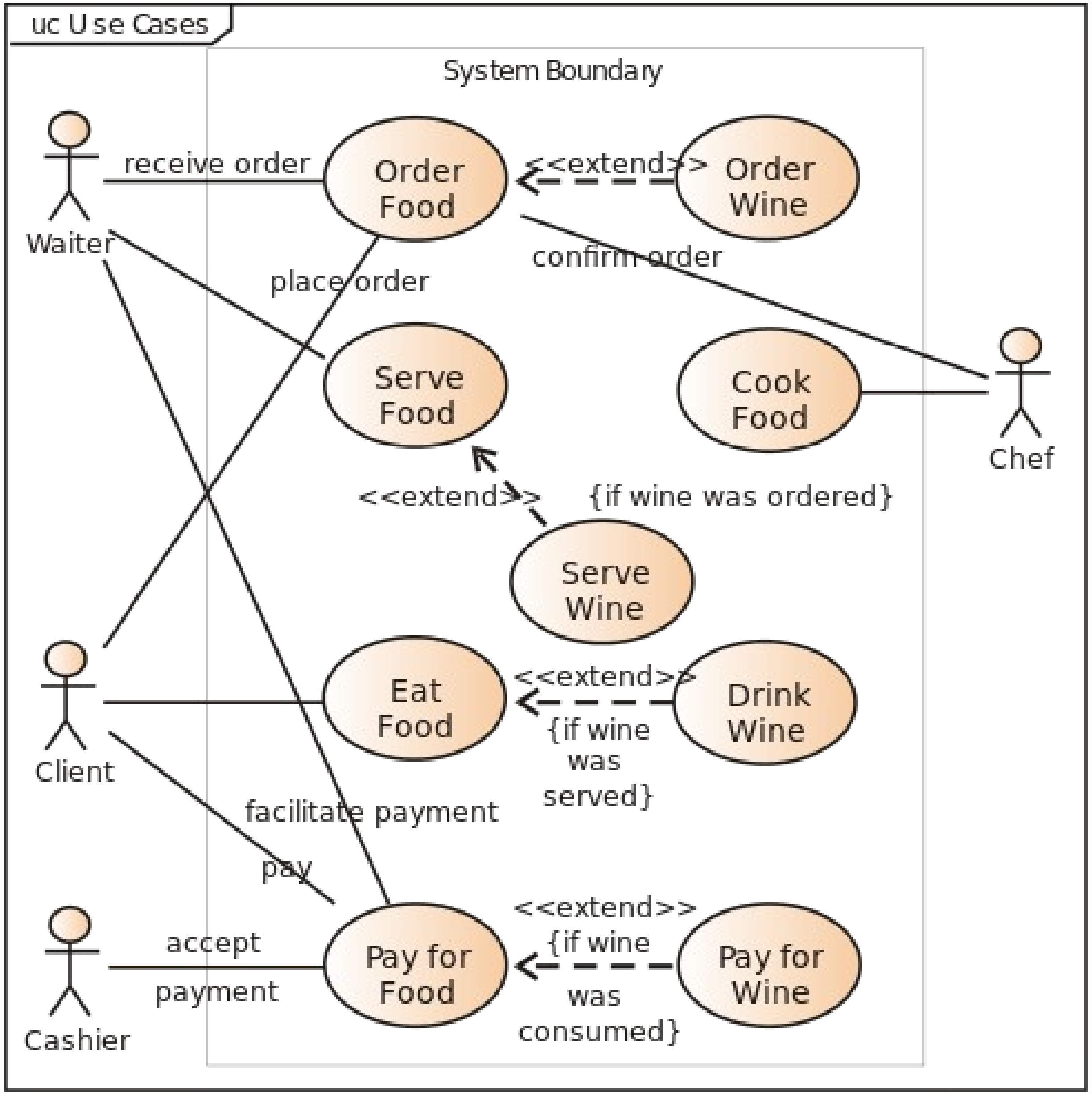
Modifiable - Having the same requirement in more than one place may not be wrong - but tends to make the document not maintainable.

Traceable - Often, this is not important in a non-politicized environment. However, in most organizations, it is sometimes useful to connect the requirements in the SRS to a higher level document. Why do we need this requirement?

sources: wikipedia, IEEE

Task 4 : Project SRS Elicitation

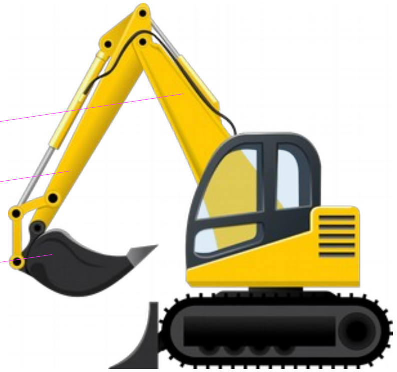
Use case: maps to W^5H to requirements (R) to specifications (S)

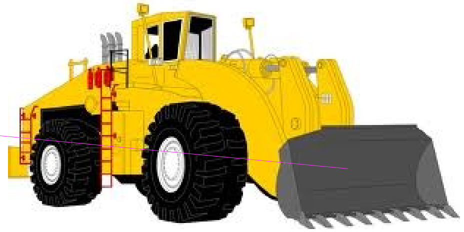


behavior

Task 4 : Project SRS Elicitation

- Given our scope from Task 2 and in lecture, consider the following elements with respect to critical thinking:
  - I operator controls: machine movement
  - II operator controls: linkage movement
  - III hydraulic cylinders
  - IV boom
  - V arm
  - VI bucket
  - VII scoop
  - VIII blade





Task 4 : Project SRS Elicitation

- Part I: Generate representative use cases
  - for each element, narrative in English how the user might create, configure, connect, and/or use it such that it generally performs one task in its entirety; e.g.,
    - XVII salad shooter
      - The user must first have defined a salad (see XVIII). Enter a command to assign it to a new salad shooter. Then enter a command to make the salad.
    - XVIII salad
      - The user must first have defined ingredients (see XIX). Enter a command to define the salad from a list of ingredients.
    - XIX ingredient
      - Enter a command to define any single ingredient as a string description.

Task 4 : Project SRS Elicitation

- Part II: Elicit questions from use cases
  - for each element ($\in\{$I..VII$\}$) enumerate set of <u>three</u> most representative W^{5}H questions, starting at 1; e.g., salad shooter (*note 8 > 3 here*):
  - XVII salad shooter
    - who

      1. who decides whether to make the salad by hand or with the appliance?
    - what:

      2. what is the mechanism to aim the salad at the receiving bowl?

      3. what is the process to activate the appliance?

      4. what is the maximum effective range for shooting cucumber slices?
    - when:

      5. when does the chef decide to activate the appliance?
    - where:

      6. where is the raw salad stored before processing?
    - why:

      7. why is the appliance effective for making a Caesar salad?
    - how:

      8. how does the appliance prevent the user from losing a finger?

Task 4 : Project SRS Elicitation

- for each of your questions (∈{who,what,when,where,why,how})
  - provide representative requirements as you think are necessary to address them
    - at least one per question and four in total per element
    - need only be reasonable and consistent, not necessarily 100% correct or optimal
    - formal procedures, terminology, and phraseology not required, but try
    - requirement states form of solution; it does not answer the question
  - cross-reference each requirement with its question number, starting with [a]
    - [a] The qualified chef must be in charge of the appliance at all times for safety. (1)
    - [b] The appliance needs a telescopic sight to see what the neighbors are cooking. (2)
    - [c] The appliance needs night-vision capability to see the neighbors at night. (2)
    - [d] The appliance needs a removable hopper for salad storage after dinner. (6)
    - [e] The appliance needs an exciting brand name to sell well. (7)
    - [f] The appliance must appeal to Stu Steiner so he will buy it. (8)

Task 4 : Project SRS Elicitation

- for each requirement
  - specify a form of the solution, starting with [A]
  - at least one per requirement and five in total per element
    - [A] The chef shall be certified according to ISO 7143 SSS v2.0. [a]
    - [B] The sight shall have an adjustable range of magnification from 8 to 10x. [b]
    - [C] The hopper dimensions shall be minimum 10(h)×10(w)×20(d) cm. [c]
    - [D] Acceptable materials are: 6061 aluminum or titanium. [d]
    - [E] The product shall be called UltraSaladShooter 9000. [e]
    - [F] The product shall be endorsed by Justin Bieber. [f]
    - [G] The product shall be endorsed by Miley Cyrus. [f]
- use your best judgment, but be consistent within the project so far and your own solution
  - somehow must address how to build, connect, and use components
  - must be reasonably consistent forward and backward: E↔UC↔Q↔R↔S
  - be useful: factually correct does not imply useful
- each level as separate section of document in order; do not nest