WP11

Specification

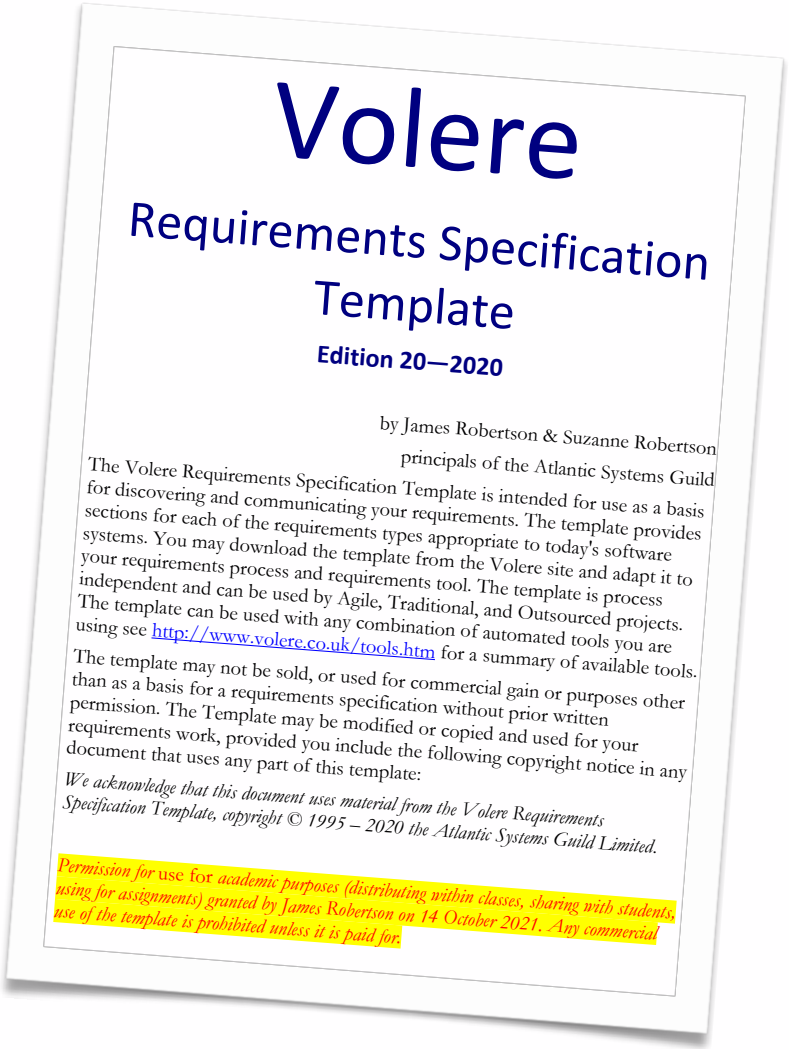
Design of Software Architectures

Dr. Vadim Zaytsev aka @grammarware, 28 September 2022

UNIVERSITY OF TWENTE.

WP10 Recap:

- Requirements fulfil the needs
  - of 1+ stakeholders
- Requirements are
  - testable / verifiable
  - realisable / approved
  - unique / atomic
  - grouped / prioritised
  - free of conflicts



Volere

Requirements Specification Template

Edition 20—2020

by James Robertson & Suzanne Robertson

principals of the Atlantic Systems Guild

The Volere Requirements Specification Template is intended for use as a basis for discovering and communicating your requirements. The template provides sections for each of the requirements types appropriate to today's software systems. You may download the template from the Volere site and adapt it to your requirements process and requirements tool. The template is process independent and can be used by Agile, Traditional, and Outsourced projects. The template can be used with any combination of automated tools you are using see http://www.volere.co.uk/tools.htm for a summary of available tools.

The template may not be sold, or used for commercial gain or purposes other than as a basis for a requirements specification without prior written permission. The Template may be modified or copied and used for your requirements work, provided you include the following copyright notice in any document that uses any part of this template:

We acknowledge that this document uses material from the Volere Requirements Specification Template, copyright © 1995 – 2020 the Atlantic Systems Guild Limited.

Permission for use for academic purposes (distributing within classes, sharing with students, using for assignments) granted by James Robertson on 14 October 2021. Any commercial use of the template is prohibited unless it is paid for.

WP08/09 Recap:

- Domain
- concepts
- Aspects
- Specification

What is a specification?

- Formal specification:

```
VARIABLE clock
Init == clock \in {0, 1}
Tick == IF clock = 0 THEN clock' = 1 ELSE clock' = 0
Spec == Init /\ [][Tick]_<<clock>>
```

https://en.wikipedia.org/wiki/TLA+

```
System simple_cs = {
  Component client = {
     Port sendRequest;
     Properties { requestRate : float = 17.0;
                  sourceCode : externalFile = "CODE-LIB/client.c" }}
  Component server = {
     Port receiveRequest;
     Properties { idempotent : boolean = true;
                  maxConcurrentClients : integer = 1;
                  multithreaded : boolean = false;
                  sourceCode : externalFile = "CODE-LIB/server.c" }}
  Connector rpc = {
     Role caller;
     Role callee;
     Properties { synchronous : boolean = true;
                  maxRoles : integer = 2;
                  protocol : WrightSpec = "..." }}
  Attachments { client.send-request to rpc.caller ;
                server.receive-request to rpc.callee }
}
```

Garlan, *Formal Modeling and Analysis of Software Architecture: Components, Connectors, and Events*, SFM 2003

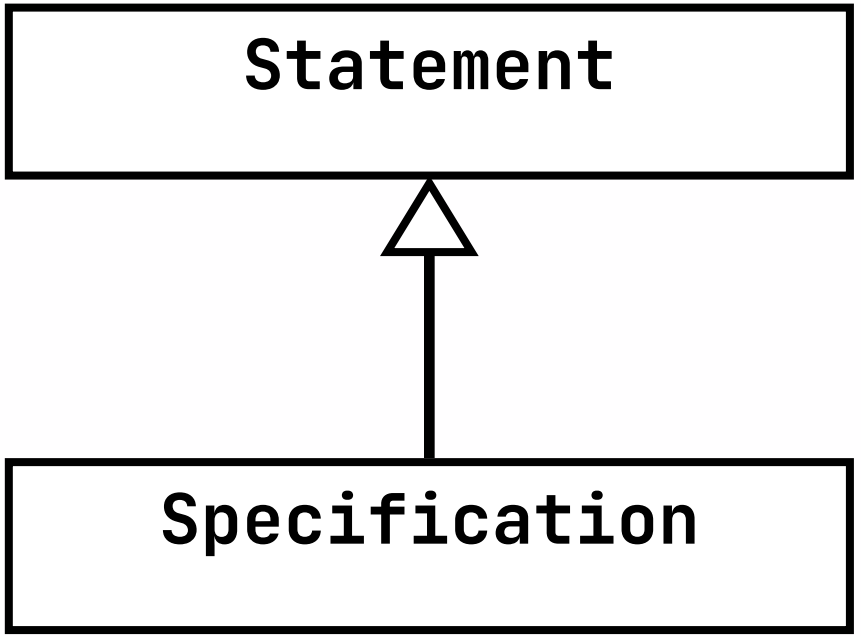
What is a specification?

- Formal specification
- “Specification” aka documentation/standard
  - requirement specification
  - functional specification
  - language/format/protocol specification
  - architecture specification

https://en.wikipedia.org/wiki/TLA+

What is a specification?

- Formal specification
- “Specification” aka documentation/standard
- Specification is a statement
  - about some aspects
  - has defined context
  - is assignable



Specification properties

- Specifications have a validity
  - wish - intention - requirement - assumption - proven
- Specifications have a scope
  - system
  - system element
  - environment element
  - range of the above
- Direct relations
  - composed of other specifications
  - derived from / based on other specifications
  - made according to other specifications
- Indirect relations via shared aspects or shared context

Specification ~ Standardisation

- Consistently specific everywhere
  - ⇝ standardised
- Wellerness: containerisation
- TurboWorkbench: plugins
- COBOL-FIT: athletic performance
- All standards you use are (external) specifications

Keep track of specifications

Keep track of all the specifications that you use and introduce

- Identification (name, version, source)
- Relation to the architecture:
  - For what architecture element is it used?
  - How does it use the architecture?
- Dependencies on other specifications
  - (graph) specifications are nodes, relations are arrows
  - (table) a row for each specification