

Documents

Marko Schütz-Schmuck

Department of Computer Science and Engineering
University of Puerto Rico at Mayagüez
Mayagüez, PR

September 7, 2020

documentation is central

- projects develop documents
 - > proposals
 - > descriptions
 - > prescriptions
 - > (stakeholder) perspectives
 - > analyses
 - > plans
 - > even the source code is a type of document

documentation is central

- projects develop documents
 - > proposals
 - > descriptions
 - > prescriptions
 - > (stakeholder) perspectives
 - > analyses
 - > plans
 - > even the source code is a type of document

documentation is central

- projects develop documents
 - > proposals
 - > descriptions
 - > prescriptions
 - > (stakeholder) perspectives
 - > analyses
 - > plans
 - > even the source code is a type of document

documentation is central

- projects develop documents
 - > proposals
 - > descriptions
 - > prescriptions
 - > (stakeholder) perspectives
 - > analyses
 - > plans
 - > even the source code is a type of document

documentation is central

- projects develop documents
 - > proposals
 - > descriptions
 - > prescriptions
 - > (stakeholder) perspectives
 - > analyses
 - > plans
 - > even the source code is a type of document

documentation is central

- projects develop documents
 - > proposals
 - > descriptions
 - > prescriptions
 - > (stakeholder) perspectives
 - > analyses
 - > plans
 - > even the source code is a type of document

documentation is central

- projects develop documents
 - > proposals
 - > descriptions
 - > prescriptions
 - > (stakeholder) perspectives
 - > analyses
 - > plans
 - > even the source code is a type of document

documentation is central

- projects develop documents
 - > proposals
 - > descriptions
 - > prescriptions
 - > (stakeholder) perspectives
 - > analyses
 - > plans
 - > even the source code is a type of document

kinds of document (parts) I

- informative
 - > concerned with meta-level
 - informing about proposed project
 - not: conducting the project
 - > can be constrained for software engineering projects
 - refer to Bjørner's SE-V3 for contents

kinds of document (parts) II

- descriptive
 - > capture physical phenomena
 - > define concepts

kinds of document (parts) III

- analytical
 - > reason over properties claimed of individual or of collections of texts
 - > examples
 - claim: "the refined requirements B1, B2, B3, ..., B23 imply the more abstract requirements A1, A2, A3, and A4"
 - method: show that if all refined requirements are met then they imply the more abstract requirements

informative documents parts or content

- name, place, date
- partners
- current situation, needs, ideas, concepts
- scope, span, and synopsis
- assumptions and dependencies
- implicit & derived goals
- standards
- contracts & design briefs
- log book

informative documents parts or content

- name, place, date
- partners
- current situation, needs, ideas, concepts
- scope, span, and synopsis
- assumptions and dependencies
- implicit & derived goals
- standards
- contracts & design briefs
- log book

informative documents parts or content

- name, place, date
- partners
- current situation, needs, ideas, concepts
- scope, span, and synopsis
- assumptions and dependencies
- implicit & derived goals
- standards
- contracts & design briefs
- log book

informative documents parts or content

- name, place, date
- partners
- current situation, needs, ideas, concepts
- scope, span, and synopsis
- assumptions and dependencies
- implicit & derived goals
- standards
- contracts & design briefs
- log book

informative documents parts or content

- name, place, date
- partners
- current situation, needs, ideas, concepts
- scope, span, and synopsis
- assumptions and dependencies
- implicit & derived goals
- standards
- contracts & design briefs
- log book

informative documents parts or content

- name, place, date
- partners
- current situation, needs, ideas, concepts
- scope, span, and synopsis
- assumptions and dependencies
- implicit & derived goals
- standards
- contracts & design briefs
- log book

informative documents parts or content

- name, place, date
- partners
- current situation, needs, ideas, concepts
- scope, span, and synopsis
- assumptions and dependencies
- implicit & derived goals
- standards
- contracts & design briefs
- log book

informative documents parts or content

- name, place, date
- partners
- current situation, needs, ideas, concepts
- scope, span, and synopsis
- assumptions and dependencies
- implicit & derived goals
- standards
- contracts & design briefs
- log book

informative documents parts or content

- name, place, date
- partners
- current situation, needs, ideas, concepts
- scope, span, and synopsis
- assumptions and dependencies
- implicit & derived goals
- standards
- contracts & design briefs
- log book

partners

- who collaborates on the project?
- includes clients, developers, additional experts

current situation, needs, ideas, concepts

- core of a proposal
- from description of the current situation we identify
- the need for some improvement and we have
- some ideas how to satisfy the needs and we use
- some concepts in this reasoning

current situation, needs, ideas, concepts

- core of a proposal
- from description of the current situation we identify
- the need for some improvement and we have
- some ideas how to satisfy the needs and we use
- some concepts in this reasoning

current situation, needs, ideas, concepts

- core of a proposal
- from description of the current situation we identify
- the need for some improvement and we have
- some ideas how to satisfy the needs and we use
- some concepts in this reasoning

current situation, needs, ideas, concepts

- core of a proposal
- from description of the current situation we identify
- the need for some improvement and we have
- some ideas how to satisfy the needs and we use
- some concepts in this reasoning

current situation, needs, ideas, concepts

- core of a proposal
- from description of the current situation we identify
- the need for some improvement and we have
- some ideas how to satisfy the needs and we use
- some concepts in this reasoning

current situation

- informal expression of the problem
- can motivate
 - > domain description
 - > requirements prescription, or
 - > software design project
 - > or combination of phases

current situation

- informal expression of the problem
- can motivate
 - > domain description
 - > requirements prescription, or
 - > software design project
 - > or combination of phases

needs, ideas

- depends on current situation
- examples
 - > need for better understanding of domain
 - > need for a system specification
 - > need for the development of a system
- ideas: how will we fulfill the needs

needs, ideas

- depends on current situation
- examples
 - > need for better understanding of domain
 - > need for a system specification
 - > need for the development of a system
- ideas: how will we fulfill the needs

needs, ideas

- depends on current situation
- examples
 - > need for better understanding of domain
 - > need for a system specification
 - > need for the development of a system
- ideas: how will we fulfill the needs

concepts and facilities

- necessary to understand the informative document
- central to the domain, to a requirements project, or to a software design

scope, span, and synopsis

- scope
 - > description of the wider area of the project
- span
 - > description of the more narrow context of the project
- synopsis
 - > summary, abstract, ...
 - > high-level view combining information from the rest of the document

scope, span, and synopsis

- scope
 - > description of the wider area of the project
- span
 - > description of the more narrow context of the project
- synopsis
 - > summary, abstract, ...
 - > high-level view combining information from the rest of the document

scope, span, and synopsis

- scope
 - > description of the wider area of the project
- span
 - > description of the more narrow context of the project
- synopsis
 - > summary, abstract, ...
 - > high-level view combining information from the rest of the document

assumptions and dependencies

- assumptions
 - > for proceeding with the project, not for using/deploying the resulting product
- dependencies
 - > in which way does the project depend on these assumptions
- example
 - > assumption
 - the client will make available domain experts in supply-chain management to consult with the developers throughout the project in a timely manner and for a sufficient duration
 - > dependency
 - if the above assumption is not met the project might become delayed due to additional work being done by developers and the resulting product might not fully meet the clients wishes

assumptions and dependencies

- assumptions
 - > for proceeding with the project, not for using/deploying the resulting product
- dependencies
 - > in which way does the project depend on these assumptions
- example
 - > assumption
 - the client will make available domain experts in supply-chain management to consult with the developers throughout the project in a timely manner and for a sufficient duration
 - > dependency
 - if the above assumption is not met the project might become delayed due to additional work being done by developers and the resulting product might not fully meet the clients wishes

assumptions and dependencies

- assumptions
 - > for proceeding with the project, not for using/deploying the resulting product
- dependencies
 - > in which way does the project depend on these assumptions
- example
 - > assumption
 - the client will make available domain experts in supply-chain management to consult with the developers throughout the project in a timely manner and for a sufficient duration
 - > dependency
 - if the above assumption is not met the project might become delayed due to additional work being done by developers and the resulting product might not fully meet the clients wishes

implicit & derived goals

- primary goal: solve the issue described in the combination of current situation, needs, ideas
- what are goals driving the primary goal?
- what "side" goals would be nice-to-haves?
- example
 - > primary goal: develop domain description, requirements prescription, and software design for supermarket cash register and stock management
 - > driving goal: reduce labor cost in maintaining inventory, so reduce operative cost
 - > "side" goal: improve in-store availability of goods by removing potential errors from restocking process

implicit & derived goals

- primary goal: solve the issue described in the combination of current situation, needs, ideas
- what are goals driving the primary goal?
- what "side" goals would be nice-to-haves?
- example
 - > primary goal: develop domain description, requirements prescription, and software design for supermarket cash register and stock management
 - > driving goal: reduce labor cost in maintaining inventory, so reduce operative cost
 - > "side" goal: improve in-store availability of goods by removing potential errors from restocking process

implicit & derived goals

- primary goal: solve the issue described in the combination of current situation, needs, ideas
- what are goals driving the primary goal?
- what "side" goals would be nice-to-haves?
- example
 - > primary goal: develop domain description, requirements prescription, and software design for supermarket cash register and stock management
 - > driving goal: reduce labor cost in maintaining inventory, so reduce operative cost
 - > "side" goal: improve in-store availability of goods by removing potential errors from restocking process

standards

- process standards
- documentation standards
- ISO, ANSI, IEEE, ITU, MIL, ...

contracts & design briefs

- contract
 - > who is party to contract?
 - > which party delivers what, when, how, in what quality, with what content?
 - > using which development principles, tools, standards, ...?
 - > what happens when non-complying?
- design brief
 - > expand on the two middle parts from contract

contracts & design briefs

- contract
 - > who is party to contract?
 - > which party delivers what, when, how, in what quality, with what content?
 - > using which development principles, tools, standards, ...?
 - > what happens when non-complying?
- design brief
 - > expand on the two middle parts from contract

log book

- milestones, meetings, key decision: acceptance, ...

descriptive document parts or content

- contains: description, prescription, specification
- types of descriptions
 - > rough sketch
 - > terminology
 - > narrative
 - > formalization

descriptive document parts or content

- contains: description, prescription, specification
- types of descriptions
 - > rough sketch
 - > terminology
 - > narrative
 - > formalization

rough sketch

- break the ice, start somewhere
- brainstorm
- does not need: consistency, completeness, uniform level of abstraction, correct definitions, conciseness, ...
- raise questions
- basis for analysis of concepts
- basis for group discussion
- trial and error
- might get scrapped

rough sketch

- break the ice, start somewhere
- brainstorm
- does not need: consistency, completeness, uniform level of abstraction, correct definitions, conciseness, ...
- raise questions
- basis for analysis of concepts
- basis for group discussion
- trial and error
- might get scrapped

rough sketch

- break the ice, start somewhere
- brainstorm
- does not need: consistency, completeness, uniform level of abstraction, correct definitions, conciseness, ...
- raise questions
- basis for analysis of concepts
- basis for group discussion
- trial and error
- might get scrapped

rough sketch

- break the ice, start somewhere
- brainstorm
- does not need: consistency, completeness, uniform level of abstraction, correct definitions, conciseness, ...
- raise questions
- basis for analysis of concepts
- basis for group discussion
- trial and error
- might get scrapped

terminology

- terms with their textual (narrative) definitions
- eventually terms in definitions need to be defined
- eventually no circular definitions
- resulting language gets used in all communications relating to the project
- can be formalized

terminology

- terms with their textual (narrative) definitions
- eventually terms in definitions need to be defined
- eventually no circular definitions
- resulting language gets used in all communications relating to the project
- can be formalized

terminology

- terms with their textual (narrative) definitions
- eventually terms in definitions need to be defined
- eventually no circular definitions
- resulting language gets used in all communications relating to the project
- can be formalized

terminology

- terms with their textual (narrative) definitions
- eventually terms in definitions need to be defined
- eventually no circular definitions
- resulting language gets used in all communications relating to the project
- can be formalized

narrative

- precise and unambiguous documentation of all relevant properties (entities, functions, events, behaviors, other concepts and phenomena)
- purpose
 - > ensure proper understanding by developers
 - > communicate with stakeholders
- best paired with formalization of same notions
- pleasing narrations
 - > good use of abstraction
 - > few concepts
 - > adequate use of complementary techniques (diagrams, formalization, source code, ...)

narrative

- precise and unambiguous documentation of all relevant properties (entities, functions, events, behaviors, other concepts and phenomena)
- purpose
 - > ensure proper understanding by developers
 - > communicate with stakeholders
- best paired with formalization of same notions
- pleasing narrations
 - > good use of abstraction
 - > few concepts
 - > adequate use of complementary techniques (diagrams, formalization, source code, ...)

narrative

- precise and unambiguous documentation of all relevant properties (entities, functions, events, behaviors, other concepts and phenomena)
- purpose
 - > ensure proper understanding by developers
 - > communicate with stakeholders
- best paired with formalization of same notions
- pleasing narrations
 - > good use of abstraction
 - > few concepts
 - > adequate use of complementary techniques (diagrams, formalization, source code, ...)

narrative

- precise and unambiguous documentation of all relevant properties (entities, functions, events, behaviors, other concepts and phenomena)
- purpose
 - > ensure proper understanding by developers
 - > communicate with stakeholders
- best paired with formalization of same notions
- pleasing narrations
 - > good use of abstraction
 - > few concepts
 - > adequate use of complementary techniques (diagrams, formalization, source code, ...)

narrative

- precise and unambiguous documentation of all relevant properties (entities, functions, events, behaviors, other concepts and phenomena)
- purpose
 - > ensure proper understanding by developers
 - > communicate with stakeholders
- best paired with formalization of same notions
- pleasing narrations
 - > good use of abstraction
 - > few concepts
 - > adequate use of complementary techniques (diagrams, formalization, source code, ...)

narrative

- precise and unambiguous documentation of all relevant properties (entities, functions, events, behaviors, other concepts and phenomena)
- purpose
 - > ensure proper understanding by developers
 - > communicate with stakeholders
- best paired with formalization of same notions
- pleasing narrations
 - > good use of abstraction
 - > few concepts
 - > adequate use of complementary techniques (diagrams, formalization, source code, ...)

formalization I

- cast knowledge to a precise language best suited for reasoning
 - > about completeness, consistency, other properties
- highest clarity of notions
- prove correctness of development
 - > example: software architecture as designed satisfies requirements
 - > example: refined requirements imply more high-level requirements
- complement other perspectives

formalization I

- cast knowledge to a precise language best suited for reasoning
 - > about completeness, consistency, other properties
- highest clarity of notions
- prove correctness of development
 - > example: software architecture as designed satisfies requirements
 - > example: refined requirements imply more high-level requirements
- complement other perspectives

formalization I

- cast knowledge to a precise language best suited for reasoning
 - > about completeness, consistency, other properties
- highest clarity of notions
- prove correctness of development
 - > example: software architecture as designed satisfies requirements
 - > example: refined requirements imply more high-level requirements
- complement other perspectives

formalization I

- cast knowledge to a precise language best suited for reasoning
 - > about completeness, consistency, other properties
- highest clarity of notions
- prove correctness of development
 - > example: software architecture as designed satisfies requirements
 - > example: refined requirements imply more high-level requirements
- complement other perspectives

formalization I

- cast knowledge to a precise language best suited for reasoning
 - > about completeness, consistency, other properties
- highest clarity of notions
- prove correctness of development
 - > example: software architecture as designed satisfies requirements
 - > example: refined requirements imply more high-level requirements
- complement other perspectives

formalization II

- do not over-formalize
- right abstraction level
- we find opportunities for abstraction in doing the formalization itself
- tools (selection)
 - > TLA+
 - > Alloytools
 - > imandra
 - > idris

formalization II

- do not over-formalize
- right abstraction level
- we find opportunities for abstraction in doing the formalization itself
- tools (selection)
 - > TLA+
 - > Alloytools
 - > imandra
 - > idris

formalization II

- do not over-formalize
- right abstraction level
- we find opportunities for abstraction in doing the formalization itself
- tools (selection)
 - > TLA+
 - > Alloytools
 - > imandra
 - > idris

formalization II

- do not over-formalize
- right abstraction level
- we find opportunities for abstraction in doing the formalization itself
- tools (selection)
 - > TLA+
 - > Alloytools
 - > imandra
 - > idris

analytic document parts or content I

- prove, designate properties of other texts or of relationships between tests
- example
 - > claim: "at the cash register customers pay for any and all items in their cart/basket"
 - > proof attempt unsuccessful: customer returns/exchanges a spoiled item
 - passes cash register with replacement item, but does not pay for it
 - > proof attempt unsuccessful: customer enters store with items from other store (placed in cart)
 - passes cash register with "foreign" items, but does not pay for them

analytic document parts or content I

- prove, designate properties of other texts or of relationships between texts
- example
 - > claim: "at the cash register customers pay for any and all items in their cart/basket"
 - > proof attempt unsuccessful: customer returns/exchanges a spoiled item
 - passes cash register with replacement item, but does not pay for it
 - > proof attempt unsuccessful: customer enters store with items from other store (placed in cart)
 - passes cash register with "foreign" items, but does not pay for them

analytic document parts or content I

- prove, designate properties of other texts or of relationships between tests
- example
 - > claim: "at the cash register customers pay for any and all items in their cart/basket"
 - > proof attempt unsuccessful: customer returns/exchanges a spoiled item
 - passes cash register with replacement item, but does not pay for it
 - > proof attempt unsuccessful: customer enters store with items from other store (placed in cart)
 - passes cash register with "foreign" items, but does not pay for them

analytic document parts or content I

- prove, designate properties of other texts or of relationships between tests
- example
 - > claim: "at the cash register customers pay for any and all items in their cart/basket"
 - > proof attempt unsuccessful: customer returns/exchanges a spoiled item
 - passes cash register with replacement item, but does not pay for it
 - > proof attempt unsuccessful: customer enters store with items from other store (placed in cart)
 - passes cash register with "foreign" items, but does not pay for them

analytic document parts or content I

- prove, designate properties of other texts or of relationships between tests
- example
 - > claim: "at the cash register customers pay for any and all items in their cart/basket"
 - > proof attempt unsuccessful: customer returns/exchanges a spoiled item
 - passes cash register with replacement item, but does not pay for it
 - > proof attempt unsuccessful: customer enters store with items from other store (placed in cart)
 - passes cash register with "foreign" items, but does not pay for them

analytic document parts or content I

- prove, designate properties of other texts or of relationships between tests
- example
 - > claim: "at the cash register customers pay for any and all items in their cart/basket"
 - > proof attempt unsuccessful: customer returns/exchanges a spoiled item
 - passes cash register with replacement item, but does not pay for it
 - > proof attempt unsuccessful: customer enters store with items from other store (placed in cart)
 - passes cash register with "foreign" items, but does not pay for them

analytic document parts or content II

- contains
 - > concept formation
 - study rough sketches
 - elaborate abstractions
 - forms base of narrative
 - > validation
 - stakeholders inspect, comment, review
 - > verification
 - includes model checking, testing
 - show properties to hold of other texts
 - example: my implementation of sqrt will return 1.0, 2.0, 3.0, 4.0, 5.0 on inputs of 1.0, 4.0, 9.0, 16.0, 25.0 respectively
 - example: the software design specification combined with the domain assumptions imply the requirements
 - > theory formation
 - build a theory of the domain

analytic document parts or content II

- contains
 - > concept formation
 - study rough sketches
 - elaborate abstractions
 - forms base of narrative
 - > validation
 - stakeholders inspect, comment, review
 - > verification
 - includes model checking, testing
 - show properties to hold of other texts
 - example: my implementation of sqrt will return 1.0, 2.0, 3.0, 4.0, 5.0 on inputs of 1.0, 4.0, 9.0, 16.0, 25.0 respectively
 - example: the software design specification combined with the domain assumptions imply the requirements
 - > theory formation
 - build a theory of the domain

analytic document parts or content II

- contains
 - > concept formation
 - study rough sketches
 - elaborate abstractions
 - forms base of narrative
 - > validation
 - stakeholders inspect, comment, review
 - > verification
 - includes model checking, testing
 - show properties to hold of other texts
 - example: my implementation of sqrt will return 1.0, 2.0, 3.0, 4.0, 5.0 on inputs of 1.0, 4.0, 9.0, 16.0, 25.0 respectively
 - example: the software design specification combined with the domain assumptions imply the requirements
 - > theory formation
 - build a theory of the domain

analytic document parts or content II

- contains
 - > concept formation
 - study rough sketches
 - elaborate abstractions
 - forms base of narrative
 - > validation
 - stakeholders inspect, comment, review
 - > verification
 - includes model checking, testing
 - show properties to hold of other texts
 - example: my implementation of sqrt will return 1.0, 2.0, 3.0, 4.0, 5.0 on inputs of 1.0, 4.0, 9.0, 16.0, 25.0 respectively
 - example: the software design specification combined with the domain assumptions imply the requirements
 - > theory formation
 - build a theory of the domain