Product Planning

3. Product Planning

3.1 Product Requirements Engineering

- Role of Requirements Engineering in Software Product Management
- Inquiry cycle with elicitation, analysis, and validation

3.2 Release Planning

- Release Planning Process and its conflicts / Structure of Release Plan

3.3 Roadmapping

- Product Roadmap and its elements
- Sources of input / Usage of Roadmaps

3.4 Product Life Cycle Management

- Phases of the Life Cycle
- Performance Management

3.5 Impact From Development Methodologies

Product Planning Approaches
Primary Application Areas

Requirements-Driven

- Legal / Regulatory
- Commodity
- Technology
- Back End
- Vendor-controlled + customer-controlled environments

Data-Analysis-Driven

- Experiments under uncertainty
- Optimization
- Innovation
- Front End
- Vendor-controlled environments

Data-Input-Driven

- AI / Machine Learning
- Data sets as input to the engine

H.-B. Kittlaus: https://www.linkedin.com/pulse/increasing-diversity-software-product-planning-hans-bernd-kittlaus/, 2020
based on Jan Bosch: HoliDev – Holistic DevOps Framework, https://www.linkedin.com/pulse/how-develop-software-jan-bosch/

Product Planning Approaches
Primary Application Areas

Requirements-Driven

- Market elicitation
- In-depth analysis
- Selection

Data-Analysis-Driven

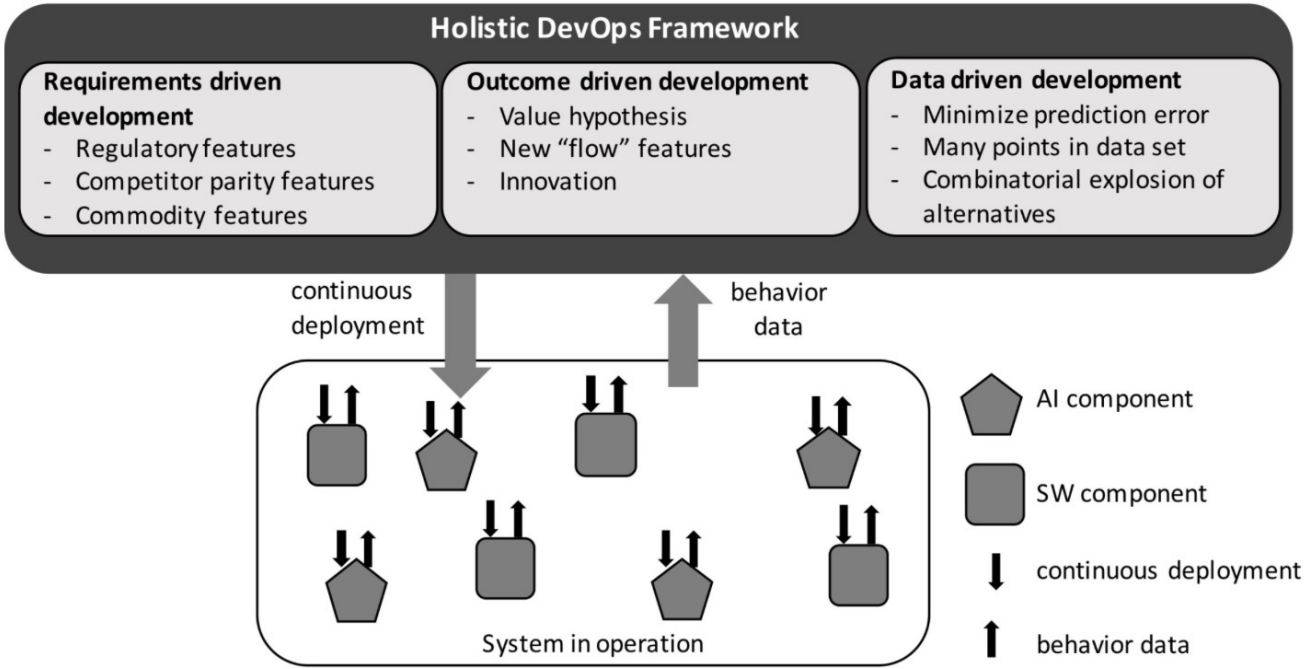
- MVP
- Experiment design + implementation, e.g.
  - A/B testing
  - Customer discovery
- Statistics
- MAB(Explore & Exploit)

Data-Input-Driven

- Data structure design
- Data collection
- Data cleansing

H.-B. Kittlaus: https://www.linkedin.com/pulse/increasing-diversity-software-product-planning-hans-bernd-kittlaus/, 2020
based on Jan Bosch: HoliDev – Holistic DevOps Framework, https://www.linkedin.com/pulse/how-develop-software-jan-bosch/

Product Planning Approaches
HoliDev Model



Jan Bosch, Professor of Software Engineering at Chalmers University of Technology

Customer Insight

Customer Insight:

Understanding of the customers‘

- problems
- environments
- jobs to be done

Not only for the product manager, but for the whole product team

➔ feed available information into discussions with stakeholders and use it for requirements analysis, as well as business modeling

Customer Insight
Customers

- Legal contractor
- Procurement Manager / Buyer
- Decision Maker / Owner
- User
- IT Manager / Operator

➔ SPM needs direct contacts

Customer Relationships: Orchestration of Marketing and Sales

Customer Insight
Direct contact activities

- Customer visits, including e.g. observation
- Meeting customers at conferences, workshops, and events
- Organizing customer round tables (focus groups)
- Design sprints with customer participation
- Supporting selected pre-sales activities
- Participating in support escalations
- Participating in online forums
- Exchange based on beta version or minimum viable product (MVP)

Customer Insight
Data analytics methods

- Monitoring online reports of market research agencies, blogs and trade press for customer information
- Monitoring, measuring, and analyzing of user behavior while they use the product
- Using data analytics software that retrieves information about customer behavior throughout the internet

Example: A/B testing

3. Product Planning
3.1 Product Requirements Engineering

What is a requirement?

- Wish for a future product feature
- Robertson & Robertson: A requirement is a statement on an action that the product is requested to do, or a quality that the product is requested to have.
- IEEE 610.12-1990: A requirement is:
  - A condition or capability needed by a user to solve a problem or achieve an objective.
  - A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
- Can also address needs and wishes outside of the software itself, e.g. sales channels, support structure, terms and conditions etc.

3. Product Planning
3.1 Product Requirements Engineering

All requirements?

- Tax rates will have to be manually entered, since the product will not interface with a tax provider program.
- The system must be able to handle 1,000 concurrent web-users per second.
- The release must be available before the new tax year.
- Menlo requires the ability to store a bill-of-lading (shipper's reference number) and a carrier pro number at the line item level for inbound (purchase) orders.
- We want the enterprise license to include our subsidiary in Dubai

3. Product Planning
3.1 Product Requirements Engineering

Three types of requirements (software-related)

- Functional requirements, which describe what the product should do.
- Quality requirements, which describe a quality that a product should have.
- Constraints, which are decisions taken in advance that restrict the scope of the product, and how the product is developed.

Sommerville (2007), Pohl (2010), Robertson & Robertson (2006), and Wiegers (2003)

3. Product Planning
3.1 Product Requirements Engineering

- Definition:
  A functional requirement is a statement of
  - a service the product should provide,
  - how the product should react to particular inputs,
  - and how the product should behave in particular situations.

- Examples:
  - “Deletion of an order will automatically delete all the lines of the order”
  - “The image viewer must display enlarged images.”

3. Product Planning
3.1 Product Requirements Engineering

- Definition:

  A quality requirement defines a quality property of the entire product or of a product component, service, or function.

- Examples:
  - "The system functions in a 7x24 mode and must have less than 1 hour downtime per month."
  - "The response time of the home page must not exceed five seconds."

- Quality requirements are sometimes referred to as non-functional requirements.

- Can be specified qualitatively, by example, operationally, or quantitatively

3. Product Planning
3.1 Product Requirements Engineering

Quality requirements

Important primarily to users

- Availability
- Efficiency
- Flexibility
- Integrity
- Performance
- Interoperability
- Reliability
- Robustness
- Usability

Important primarily to developers

- Scalability
- Maintainability
- Portability
- Reusability
- Testability

Wiegers (2003)

Quality requirements (users)

- Availability, concerning the percentage of time that a product is available for use and fully operational.
- Efficiency, referring to how efficient the product is in using resources as processor time, memory, or communication band with.
- Flexibility, which indicates how easily a product can be extended with new functionalities.
- Integrity, concerning protection against unauthorized access, data privacy, information loss, and infections through maleficent software.
  - Security (no intentional harm)
  - Safety (no harm)
- Interoperability, referring to how easily the product can exchange data or services with other systems.

Quality requirements (users)

- Reliability, indicating how long a product can be used without failure.
- Robustness, which is the degree to which the product or product component continues to operate correctly when confronted with invalid inputs, defects in connected systems, or unexpected operating conditions
- Usability, which refers to the effort that is needed of the user to prepare input for, operate, and interpret the output of the product.

Quality requirements (developers)

- Scalability, refers to the range of workload scenarios in which the software can run with satisfying performance
- Maintainability, which indicates the effort it takes to correct a defect or make a change in the product.
- Portability, indicating how easy it is to migrate a product or product component from one operating environment to the other.
- Reusability, referring to the extent to which a product component can be reused in other products.
- Testability, which indicates the effort it takes to test the product (components) to find defects.

Examples for quality requirements

- Performance
  “The system must be able to handle 1,000 concurrent web-users per second”
- Integrity
  “The parameter setting of the system can only be entered and modified by a user with super-user rights”
- Availability
  “The system functions in a 7x24 mode"
- Reliability
  “The system must have less than 1 hour downtime per month"

Handling quality requirements

- Quality requirements should not be “hidden” in a functional requirement. They are described:
  - Inside a functional requirement in case it is directly related
  - A separate requirement in case it is unrelated and requires significant workload.
- Functional requirements should not be “hidden” in a quality requirement.
- Quality requirements must have a business case, and are not for the sake of the Development team.

Example of an NFR / underspecified functional requirement

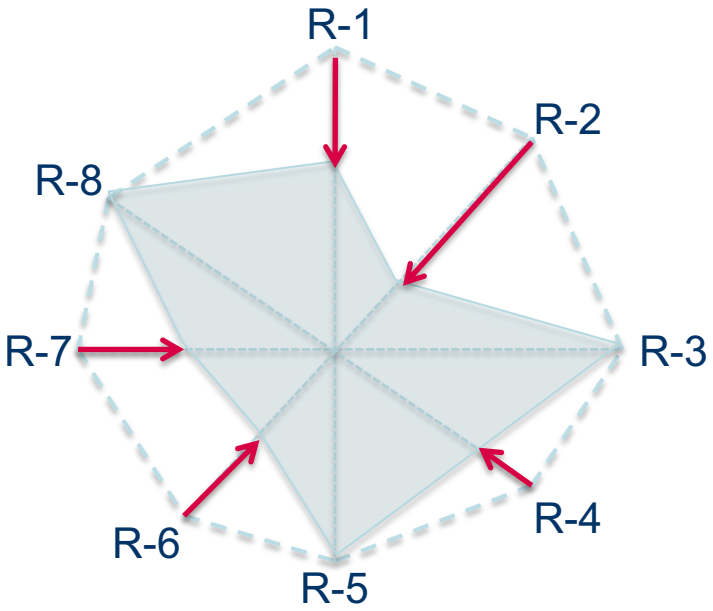
- "The system shall be secure."
  - What does "secure" mean?
  - Which properties should it have to be "secure"?
  - How can one check whether the system is "secure"?
- Breakdown of the NFR:
  - Each user must log in to the system with his user name and password prior to using the system. (functional requirement)
  - The system shall remind the user every four weeks to change the password (functional requirement)
  - When the user creates or changes the password, the system shall validate the new password is at least eight characters long and contain alphanumeric characters. (functional requirement)
  - The user passwords stored in the system must be protected against password theft (quality requirement – integrity)

3. Product Planning
3.1 Product Requirements Engineering

- Definition:
  A constraint is an organizational or technological requirement that restricts the way in which the system shall be developed.

- Examples:
  - "The product shall only operate on a smartphone."
  - "The product shall be released before the new tax year."

Restricting effects of constraints



Range of realisation alternatives for requirement without considering constraints

Range of possible realisation alternatives with the consideration of constraints

Restricting effect of constraints on a requirement

3. Product Planning
3.1 Product Requirements Engineering

Stakeholders as sources of requirements

- User groups
- Customers
- Partners
- Consultants/Professional Services
- Competitive Analysis
- Market research
- Research
- Development
- Sales
- Marketing
- Support
- Executive Management

3. Product Planning
3.1 Product Requirements Engineering

- In software product management, it is necessary to distinguish customer requirements and product requirements

Customer requirements

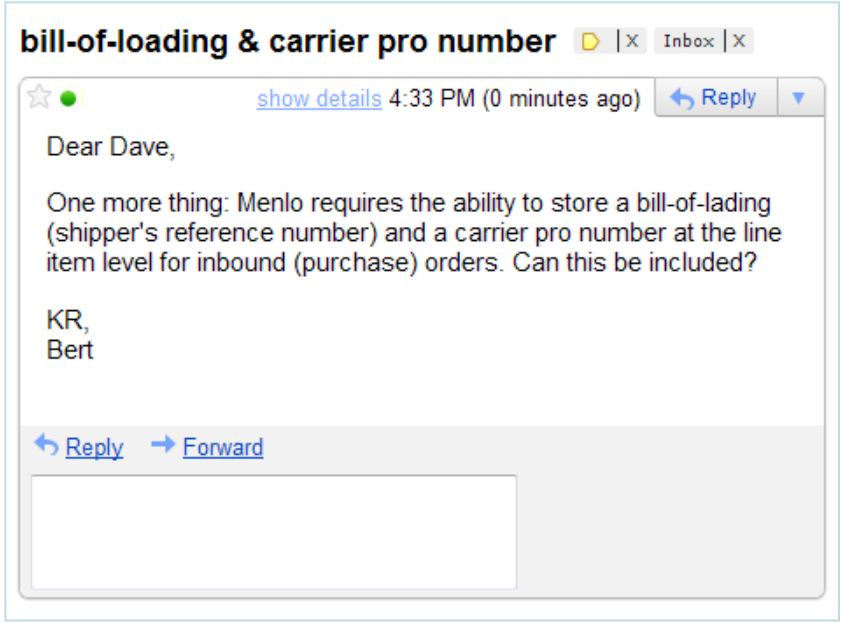
- Varying quality, vague
- Varying size: too small, too large
- Combine several wishes
- Non-standard, customer specific wishes
- Often goal-oriented

Product requirements

- Similar style suitable for internal communication
- Oriented towards the standard product

3. Product Planning
3.1 Product Requirements Engineering

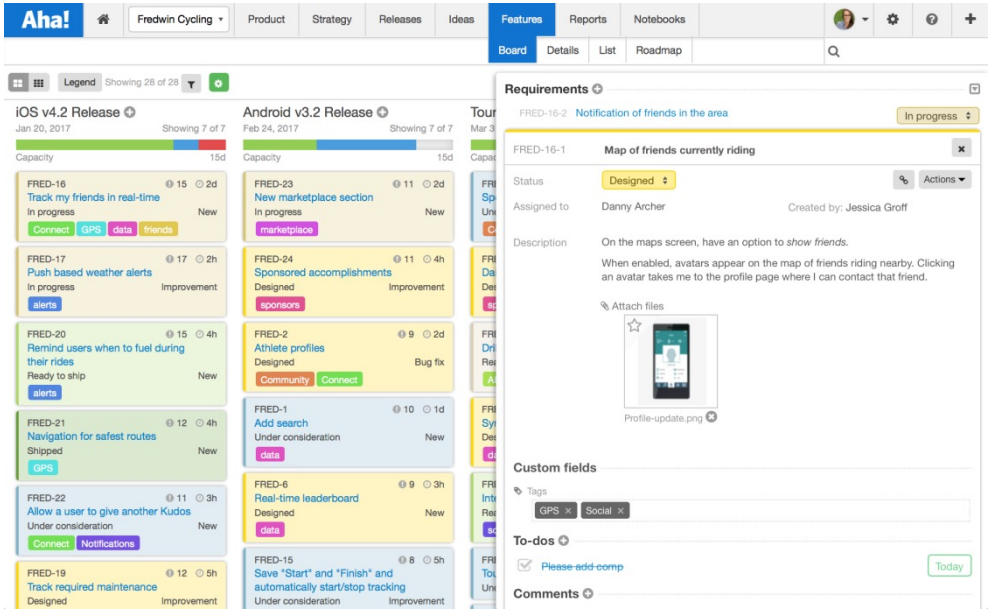
- A customer requirement is a customer wish related to current or future needs, defined using the terminology and context of the customer.



Consignment stocks of new parts are managed by DOC and consists of parts with and without serial numbers. Management of government owned stock with associated documents. Querying, access, inventor

3. Product Planning
3.1 Product Requirements Engineering

- A product requirement is a requirement to be covered by future product releases described in the company's own terminology and context.
- It typically addresses the needs of a larger group of (potential) customers.



3. Product Planning
3.1 Product Requirements Engineering

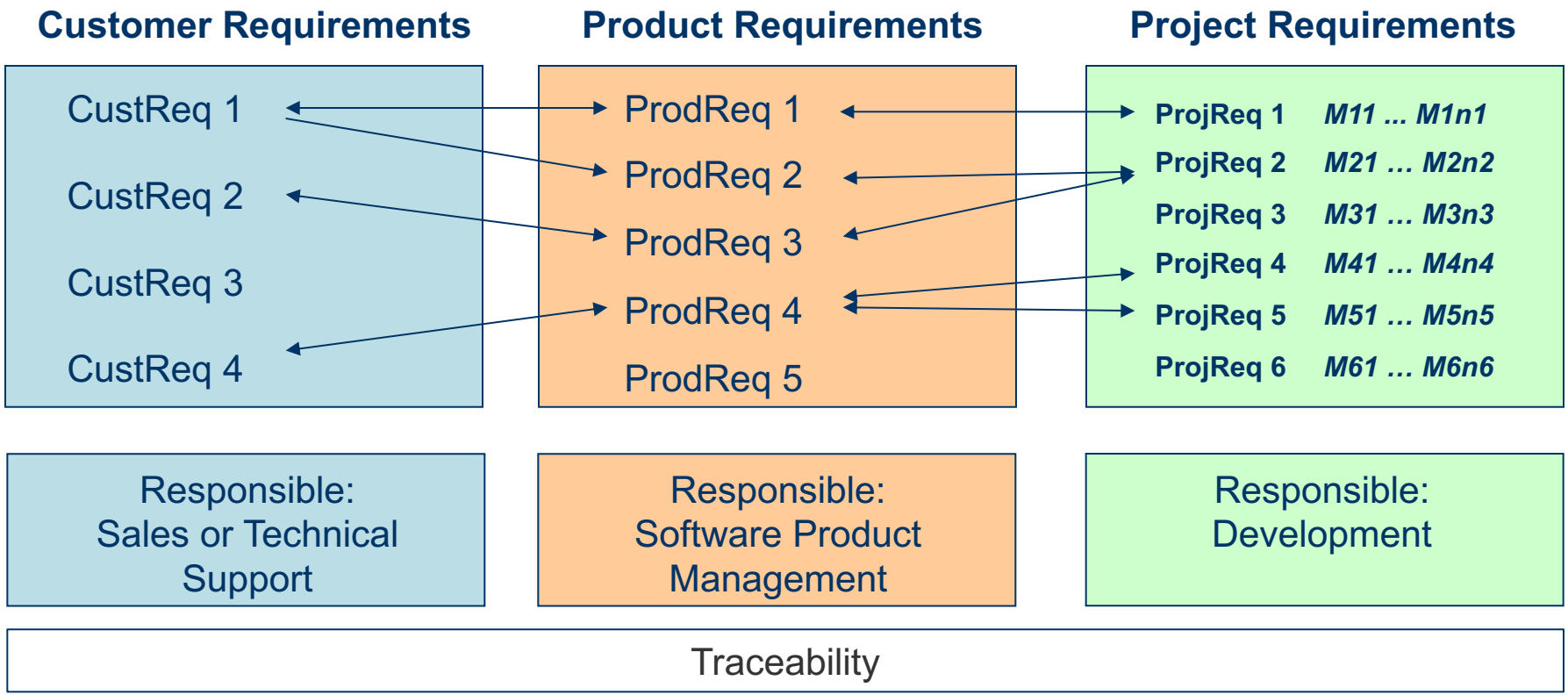
- Product requirements are usually managed and maintained in appropriate tool environments.
- With waterfall approaches, a formal Market Requirements Document can be used, e.g.

1.0	Strategy and Overview (subset of product strategy)	
	1.1	Goals and Objectives.
	1.2	Strategic Road Map
	1.3	External Positioning
	1.4	Market Segments and Customer Categories (User Profiles)
	1.5	Value Proposition
	1.6	Value Chain Structure
	1.7	Competitive Strategy incl. Strengths and Weaknesses
2.0	Bill of Materials (product deliverables)	
3.0	Requirements	
	3.1	Internally Committed Requirements
	3.2	Externally Committed Requirements
	3.3	Highly Desirable Requirements
	3.4	Future Requirements
	3.5	Features Not Being Implemented

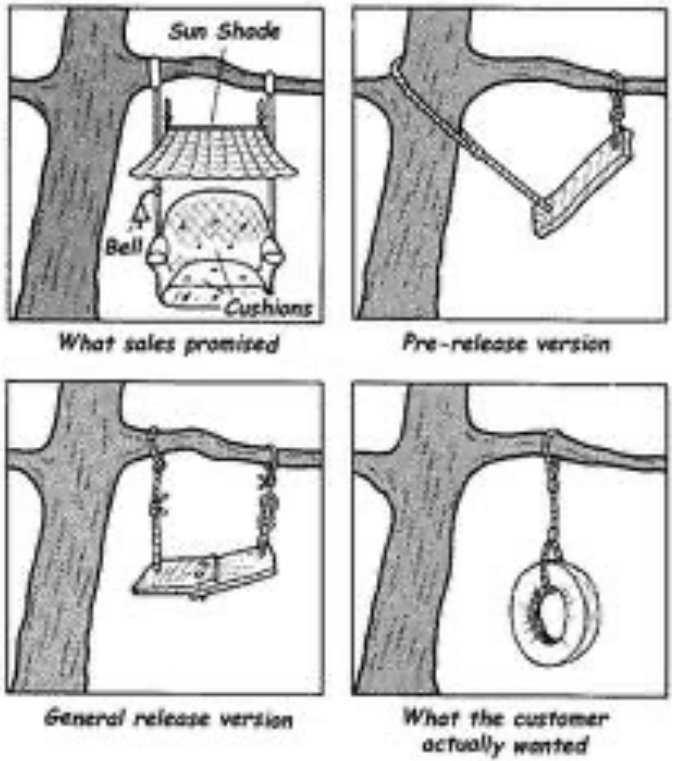
What about project requirements?

- Only related to the software developed in the project
- Often during requirements management project issues or project requirements appear: issues that apply to the project that builds the product.
  - Project issues clearly do not define the product
- Not the task of the product manager, but taken care of by the development manager or project manager.

3. Product Planning
3.1 Product Requirements Engineering



3. Product Planning
3.1 Product Requirements Engineering



Quality criteria for product requirements (1)

- Complete: The product requirement is complete when it adheres to the rules and guidelines and it does not omit any information that is relevant for any of the stakeholders.
- Traceable: The source, evolution and impact and use in later development phases should be registered.
- Correct: The relevant stakeholders should confirm its correctness and demand that the product must realize the requirements completely. A requirement is incorrect when it unnecessarily adds functionality or quality properties.
- Unambiguous: The requirement should be written in such a way that it permits only one valid interpretation.

Pohl (2010)

Quality criteria for product requirements (2)

- Comprehensible: The requirement is comprehensible if the content is understood by all relevant stakeholders.
- Consistent: The statements in the requirement should not contradict with each other.
- Verifiable: The stakeholders should be able to check whether the realized product fulfils the documented requirements or not. Acceptance criteria can be defined to facilitate verifiability.
- Up-to-date: A requirements is up-to-date when it reflects the current status of the product and product context, such as current legal regulations.

Pohl (2010)

3. Product Planning
3.1 Product Requirements Engineering

Epics and User Story Specification Template:

As a <type of user>

I want <some goal or objective >

So that <benefit, value>

Ambiguous terms to avoid in product requirements

- Acceptable, adequate (what constitutes acceptability?)
- As much as practical (don't leave this up to the developers)
- Depends on (describe the nature of the dependency)
- Efficient, fast (quantify)
- Flexible (to what?)
- Ideally (also describe non-ideal behaviour)
- Optionally (define whether this is a system, user or developer choice)
- Several (how much?)
- Shouldn't (try to state requirements as positives)

Wiegers (2003)

3. Product Planning
3.1 Product Requirements Engineering

RE Constraints		
Total	419	100%
Shared Understanding	214	51%
Specification Quality	197	47%
Clear Scope	160	38%
Efficiency	155	37%
User Satisfaction	145	35%
Timeliness	139	33%
Fit of Solution	94	22%
Estimation Reliability	65	16%
Architecture Quality	58	14%
Cost/Benefit Analysis	26	6%
Other	4	1%

Fricker (2012): Requirements Engineering State-of-Practice. Blekinge Institute of Technology.

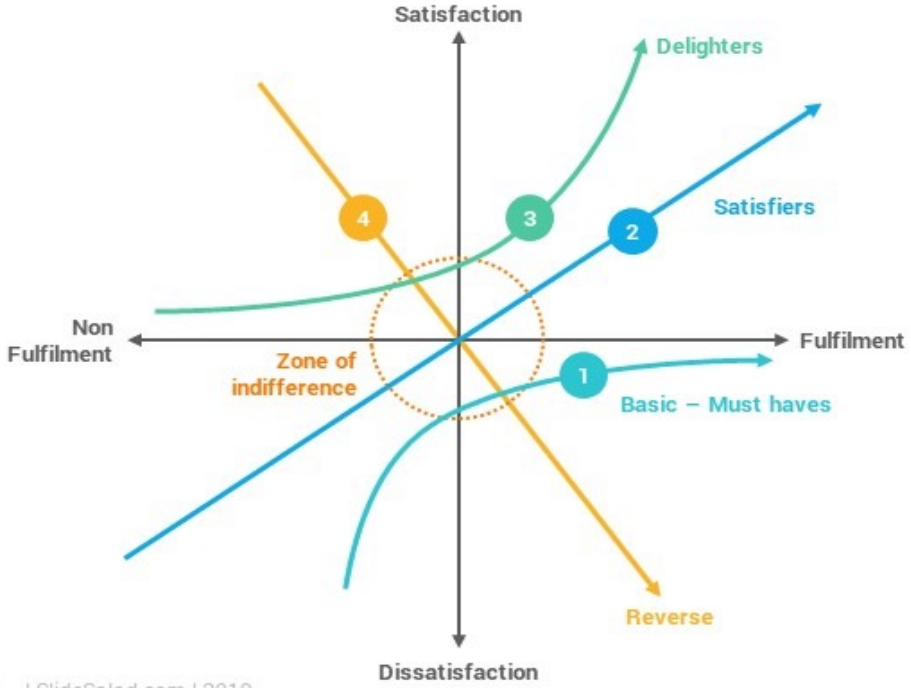
3. Product Planning
3.2 Software Product Scenarios

Software Product Scenarios		Life Cycle Phase	
		New Product Revolution	Existing Product Evolution
Runtime Environment	Vendor-Controlled	Powerboat	Speedboat
	Customer-Controlled	Icebreaker	Cruiseship

H.-B. Kittlaus (2015): One Size Does Not Fit All: Software Product Management For Speedboats vs. Cruiseships, in: Fernandes, J.M., Machado, R.J., Wnuk, K. (Eds.): Software Business, Proceedings of ICSOB 2015, Braga, Portugal, Springer

Kano Model Analysis Template

The Four Customer Reactions by Requirement Type



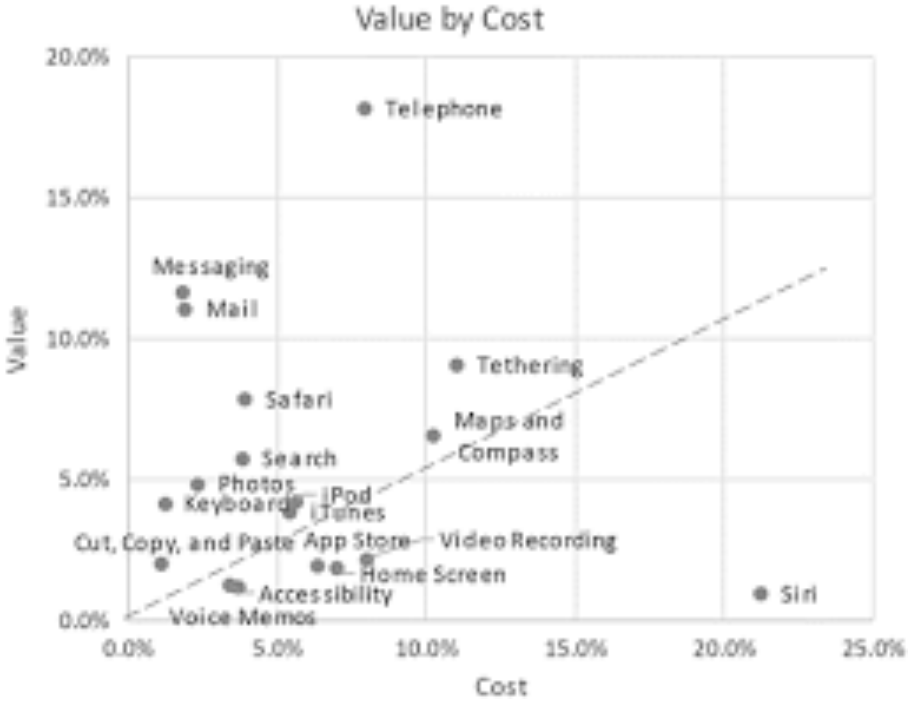
Dissatisfiers (Must Haves):
These requirement can dissatisfy a customer (if they are absent) but cannot increase satisfaction by their presence.

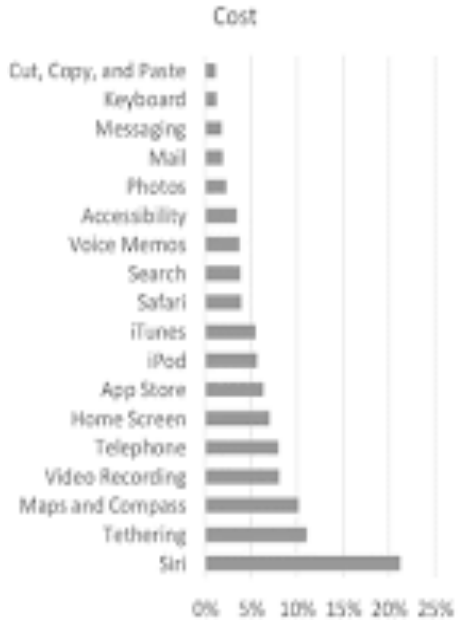
Satisfiers:
The more of these requirement that are met, the more the customer is satisfied.

Delighters:
If the requirement is absent it does not cause dissatisfaction, but it will delight the customer if it is present.

Reversers:
There features or attributes actually cause customer frustration or dissatisfaction

3. Product Planning
3.1 Product Requirements Engineering





3. Product Planning
3.1 Product Requirements Engineering

Workshops create an
efficient, controlled, and dynamic setting for quickly eliciting, prioritizing, and agreeing on requirements.

Fricker (2012): Requirements Engineering State-of-Practice. Blekinge Institute of Technology.

E. Gottesdiener (2002): Requirements by Collaboration. Addison-Wesley.

Recommendations

- Customer requirements are usually taken from customers, prospects, analysts on an as-is basis, often by Sales.
- Product requirements are formulated by the product managers.
- Communication is key
- Very small requirements, e.g. ‘extension of field length’, are entered under one generic PR per area

Managing requirements is complex

- Large volumes that have to be handled
- Complex dependencies between the requirements
- Involvement of a diversity of stakeholders
- Decisions need to be based on extensive domain knowledge of the (industrial) applications of the product.

Typical Situation in a Market-Driven Environment

>>5‘000 vague needs and ideas

500-5‘000 requirement candidates

10-500 concrete requirements
(to be implemented in software)

Gorschek, Wohlin: „Requirements Abstraction Model“, Requirements Engineering, 2006.
Regnell, Svensson, Wnuk: „Can We Beat the Complexity of Very Large-Scale Requirements Engineering?“, RefsQ‘08.

Process and Tools

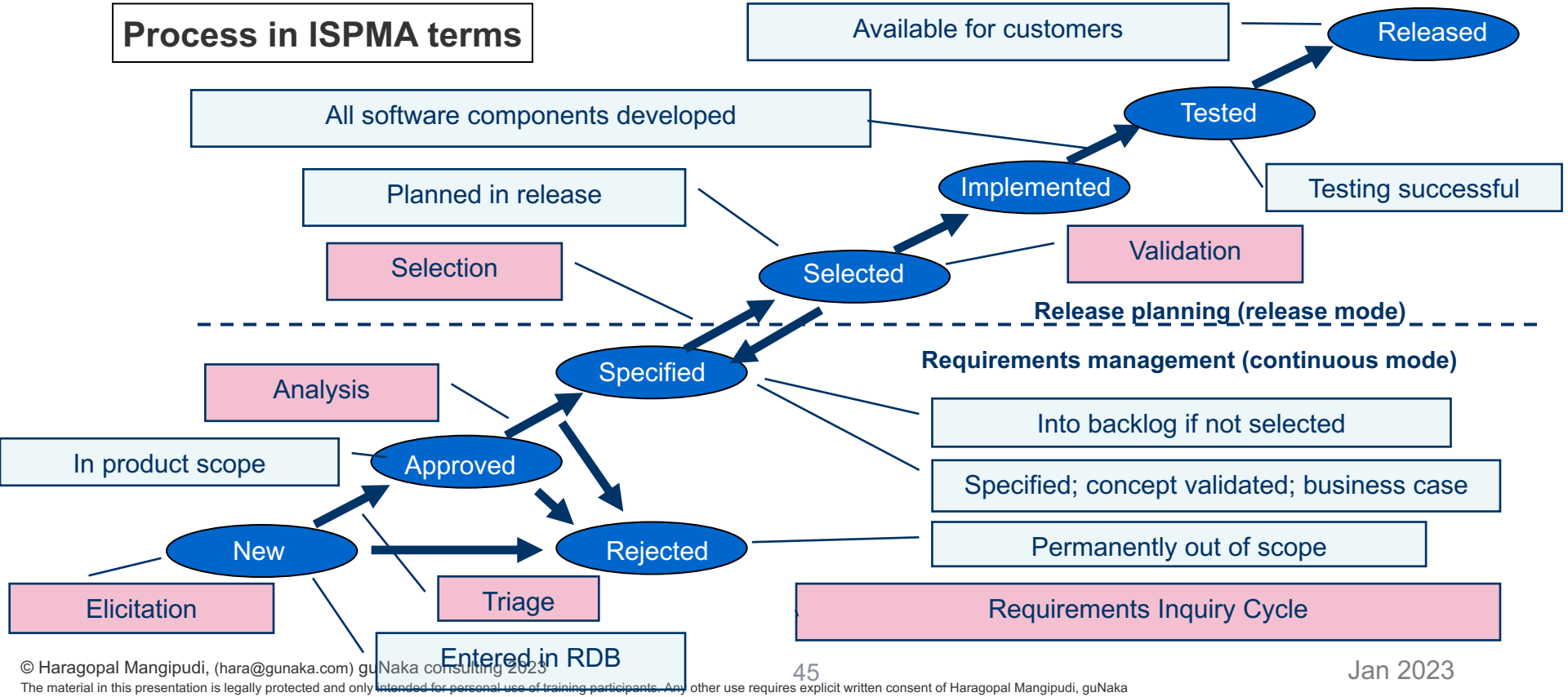
- Requirements management process needed
  - Well defined, documented and communicated to all stakeholders
  - SPM as process owner

- Tools are available and helpful
  - Requirements management tool
  - Requirements database

Product requirement‘s attributes

Attribute	*Value*	*Assigned in State*
State	N / A / Sp / Se / D / T / Rel / Rej	-
Name	Short unique name	New
ID	Unique identifier	New
Source	Who issued it?	New
Description	Short textual description	New
Func Component	Affected (sub-)modules	N / A / Sp
Priority	Importance category (1, 2, 3)	Specified
Motivation	Rationale: Why is it important	Specified
Specification	Links to Use case, Conc. Solution	Specified
Links	Links to other reqs; parent-child rel.	Specified
Estimation	Effort estimation in hours, benefit estimation	Specified
Schedule	Selected for this release	Selected
Design	Links to design documents	Implemented
Test	Links to test documents	Tested
Release Ver	Released in this version	Released

3. Product Planning
3.1 Product Requirements Engineering



3. Product Planning
3.1 Product Requirements Engineering

Requirements Inquiry Cycle: Elicitation

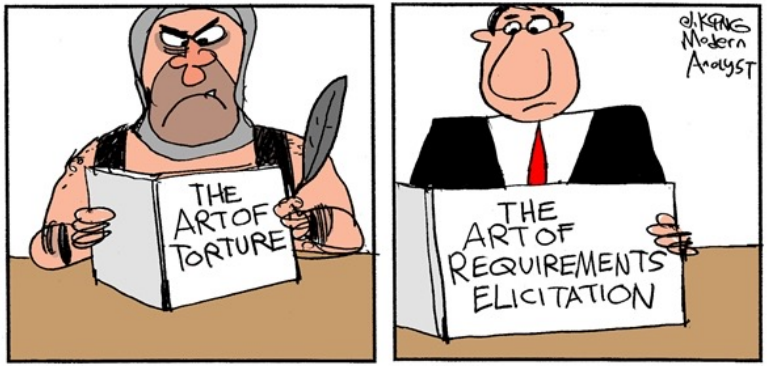
- Interviews
  - Structured and unstructured interviews
- Workshops
  - Creativity techniques: brainstorming, 6 thinking hats, ladders
  - Analysis techniques: card sorting, repertory grids, mind-mapping
  - Design techniques: prototyping (e.g. Joint Application Design, JAD)
  - Feedback techniques: reviews, focus groups
- Observation
  - Protocol analysis
  - Apprenticing
  - Ethnography

Pohl (2007): Requirements Engineering. d.punkt Verlag. AND
Zowghi, Coulin (2005):"Requirements Elicitation: A Survey of Techniques, Approaches, and Tools". In Aurum, Wohlin: Engineering and Managing Software Requirements. Springer.

3. Product Planning
3.1 Product Requirements Engineering

Requirements Inquiry Cycle: Elicitation

- Surveys
  - Questionnaires, self-recording
- Introspection
- Artefacts
  - Perspective-oriented reading
  - System archeology



Job titles have evolved, techniques not so much...

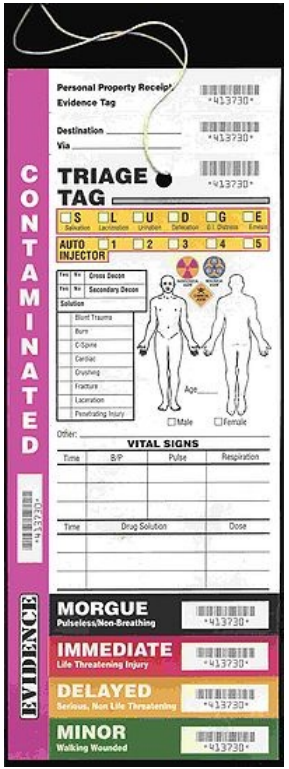
Pohl (2007): Requirements Engineering. d.punkt Verlag. AND
Zowghi, Coulin (2005):"Requirements Elicitation: A Survey of Techniques, Approaches, and Tools". In Aurum, Wohlin: Engineering and Managing Software Requirements. Springer.

3. Product Planning
3.1 Product Requirements Engineering

Requirements Inquiry Cycle: Triage

- Before applying prioritization techniques: perform triage
  - Origins in medicine

- Some requirements must be included
- Some requirements should definitely be excluded
- That leaves a pool of nice-to-haves, which need in-depth analysis.



Requirements Inquiry Cycle: Analysis

- Gather and prepare all information that is needed for decision making
  - What is it?
    - Develop a deep understanding of requirement and relevant scenarios
    - Describe it in short text
  - How can it be implemented?
    - Concept
    - High-level specification (e.g. text, use case, model, prototype)
- With agile development methodologies: a larger part of analysis is done as part of development

Requirement analysis: business case

- Business value / benefit
  - Absolute values (“How much extra money will we earn when we implement this requirement?”)
  - Relative values (“Requirement A will generate two times as much revenue as requirement B.”)

- Penalty if not developed / harm avoidance
  - Absolute values (“How much money will we lose due to decreasing sales if we do not make a web version of our product?”)
  - Relative values (“The harm done will be higher if we leave out the requirements submitted by customer C than customer D.”)

Requirement analysis: business case

- Cost of development
  - In € or $
  - In person days
  - Relative (“Requirements 1 costs twice as much as requirement 2”)

- Development risk
  - Requirement volatility / stability (“Is the requirements likely to change?”)
  - Development difficulty (“This requirement concerns a new technology, which our developers have never used before.”)

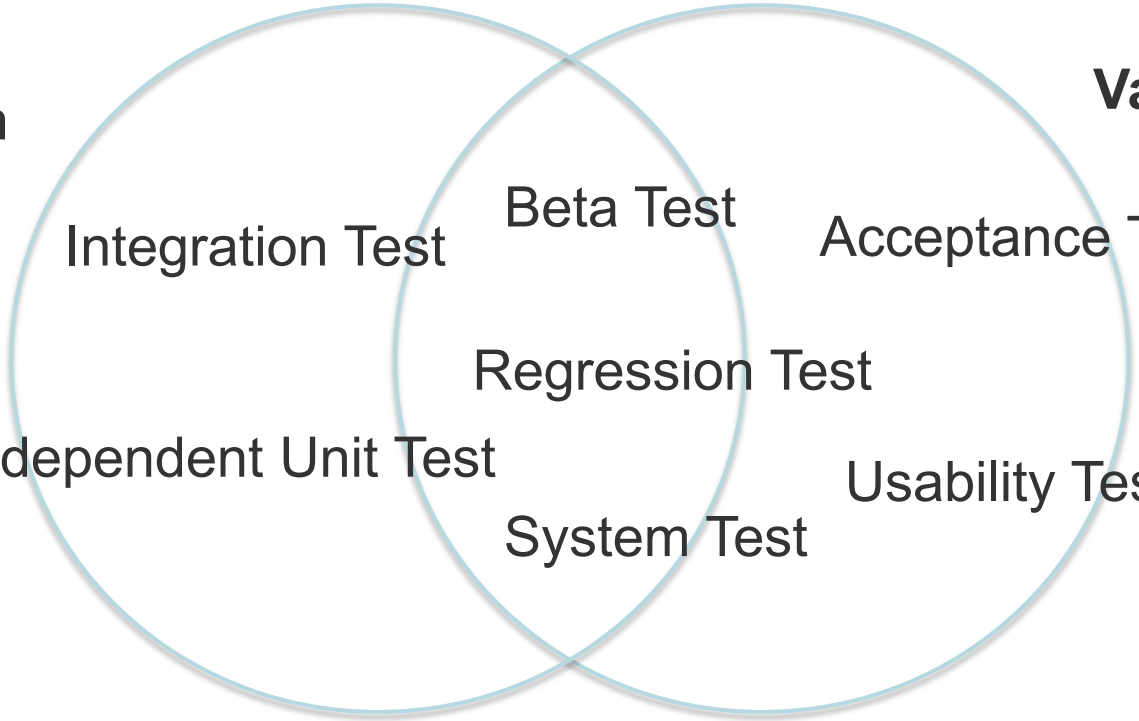
Requirements Inquiry Cycle: Selection

- Decision process for the contents of a particular release
- Decision based on
  - Evaluation
  - Release themes
  - Prioritization
- Part of Release Planning process

Requirements Inquiry Cycle: Validation

- Ensure that the specified solution is adequate and gets stakeholder acceptance
- Methods:
  - Reviews
  - Inspections
  - Simulations
  - Prototyping

3. Product Planning
Verification and Validation



Are we building the product right ?

Are we building the right product?

Agenda

3. Product Planning

3.1 Product Requirements Engineering

- Role of Requirements Engineering in Software Product Management
- Inquiry cycle with elicitation, analysis, and validation

3.2 Release Planning

- Release Planning Process and its conflicts / Structure of Release Plan

3.3 Roadmapping

- Product Roadmap and its elements
- Sources of input / Usage of Roadmaps

3.4 Product Life Cycle Management

- Phases of the Life Cycle
- Performance Management

3.5 Impact From Development Methodologies

3. Product Planning
3.2 Release Planning: Types of Versioning

Software Product Scenarios		Life Cycle Phase	
		New Product Revolution	Existing Product Evolution
Runtime Environment	Vendor-Controlled	Powerboat	Speedboat
	Customer-Controlled	Icebreaker	Cruiseship

H.-B. Kittlaus (2015): One Size Does Not Fit All: Software Product Management For Speedboats vs. Cruiseships, in: Fernandes, J.M., Machado, R.J., Wnuk, K. (Eds.): Software Business, Proceedings of ICSOB 2015, Braga, Portugal, Springer

3. Product Planning
3.2 Release Planning: Types of Versioning

Powerboat + Speedboat

- High release frequency, smaller releases
- Special methods and techniques, e.g.:
  - DevOps
  - Product Discovery

Icebreaker + Cruise Ship

- Medium to low release frequency, bigger releases
- More elaborate planning

Evolution of software (intensive) products

Original iPhone

iPhone 4 + 4S

iPhone 6S (Plus)

iPhone 8 (Plus)

iPhone 11 (Plus)

iPhone 3G

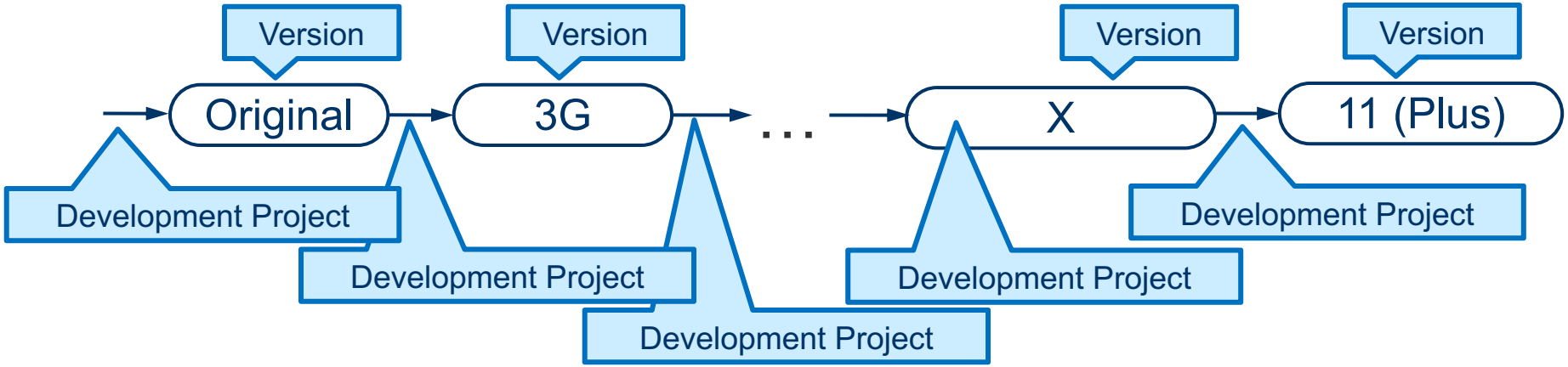
iPhone 5S + 5C

iPhone 7 (Plus)

iPhone X

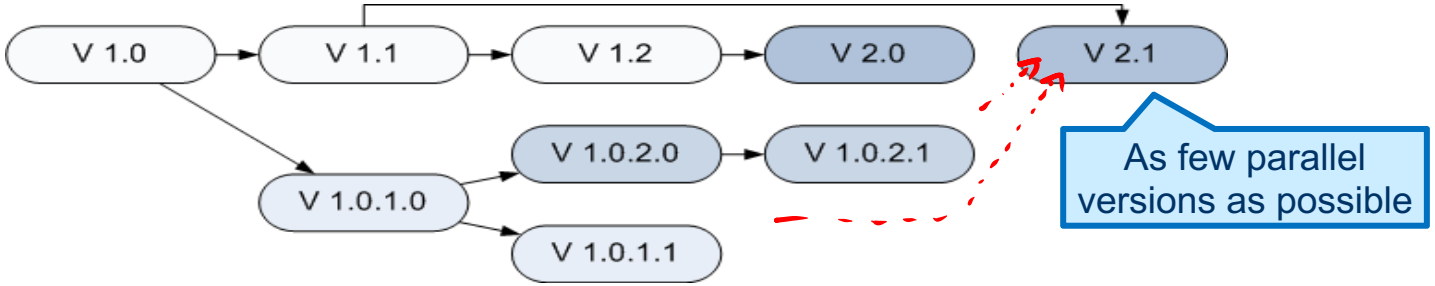
source: apple

Evolution of software (intensive) products



Evolution of software (intensive) products

- A software product exists in different versions during its lifecycle.
- Conventions:
  - „X.0“ = significant changes,
    „X.Y“ = improvements
  - Alternative: publication year (e.g. Windows 2007)



3. Product Planning
3.2 Release Planning

- Defining the contents of a release by
  - Selecting the optimal set of requirements to be implemented
  - Documenting the release contents
  - Validating the results of Development

- **Overall objective**:
  Creation of additional **value** to customers that can be transformed into **economic success** over the product‘s lifecycle

3. Product Planning
3.2 Release Planning

- Complex iterative process to find a balance between conflicting objectives:
  - Technology push vs. market pull, i.e. innovation vs. customer requirements
  - Thematic theme(s)
  - Prioritized requirements (ideally based on req. business cases)
  - Target cost
  - Release business case (ROI)
  - Complex dependencies (thematic, technical, temporal)
  - Competitive situation
  - Customer commitments
  - Marketing events (fairs etc.)

- Tightly linked to Requirements Engineering

3. Product Planning
3.2 Release Planning: Release Types

Release type	Contents	Issues
Major	Significant changes, e.g. new platform, new technology, new user interface etc.	Often as a new version (within temporal versioning) for marketing reasons or price impact
Minor	Functional changes of medium magnitude	
Update	Small functional changes + error correction	Consumer market: Often replaced by automatic online updates; Enterprise market: Will typically only be installed by customers who have one of the fixed problems
Service (Patch)	Only error correction	Consumer market: Often replaced by automatic online updates; Enterprise market: Will typically only be installed by customers who have one of the fixed problems

Heartbeat principle

- Implement a corporate release heartbeat

- Advantages are:
  - Clarity for defining a release agenda
  - Professional internal atmosphere
  - Professional image
  - Healthy pressure
  - Managed customer expectations

3. Product Planning
3.2 Release Planning

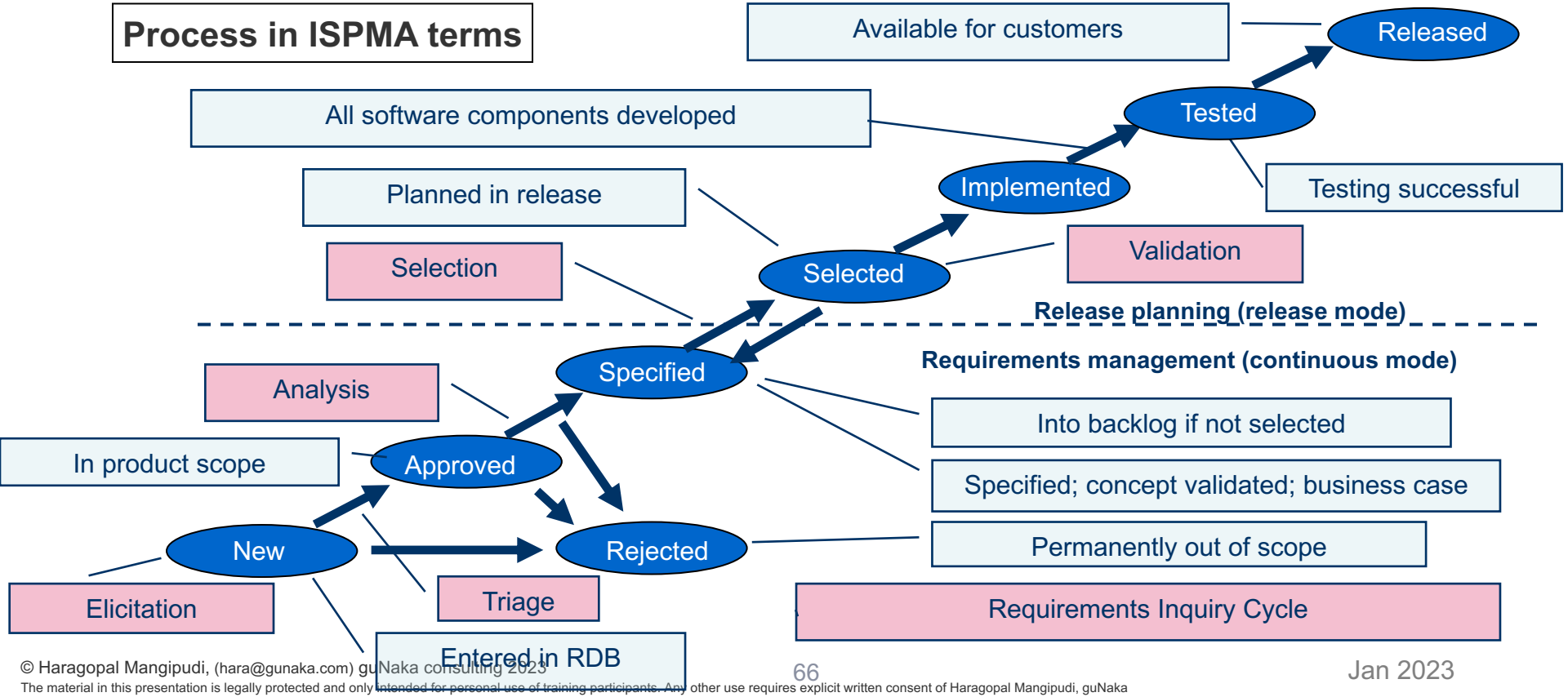
Fundamental decision: Version/Release Compatibility

- Upward compatibility
  - Existing functions of version n continue to be supported in version n+1
  - Data from version n can be transferred and used in version n+1 without changes
  - Interfaces of version n remain unchanged

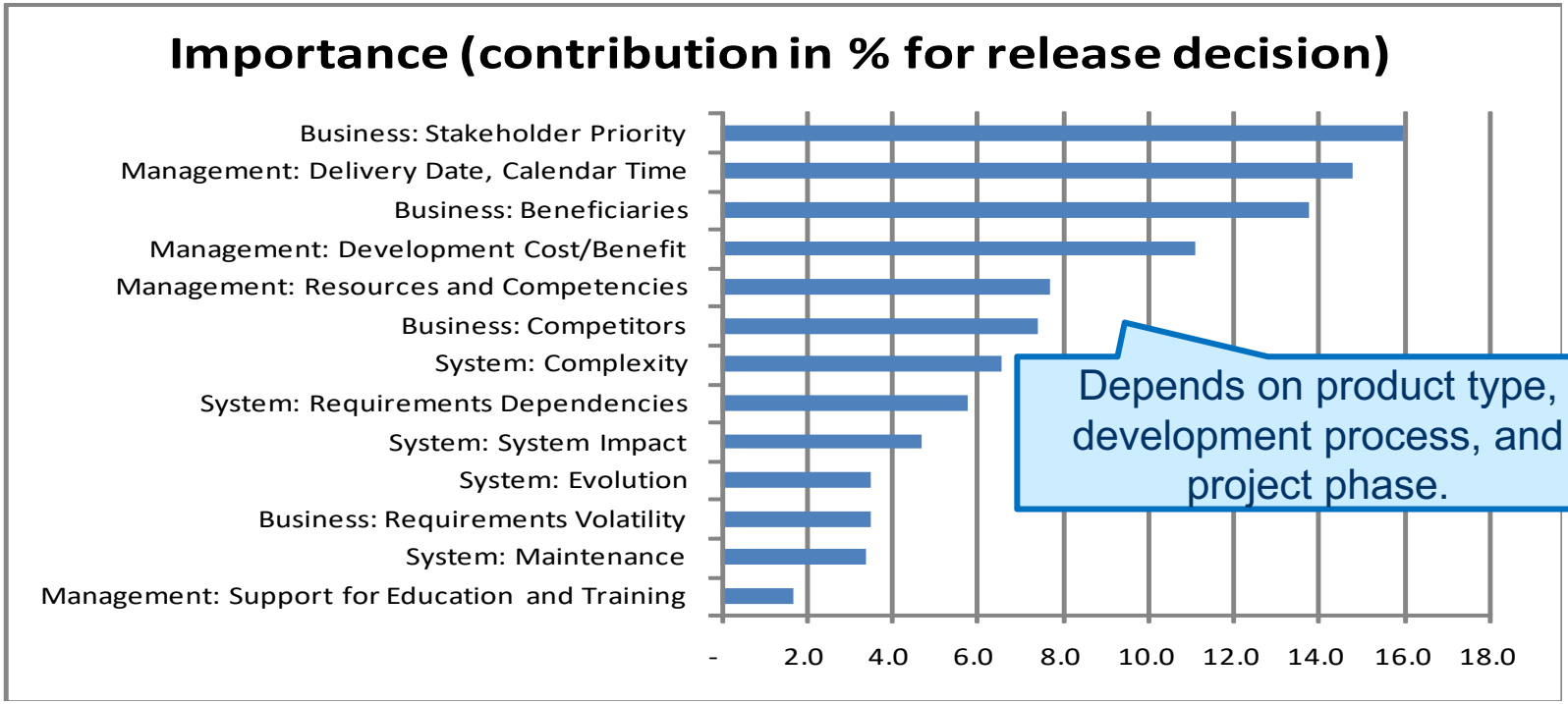
- Downward compatibility
  - Data from version n+1 can be transferred to version n without changes
  - Version n+1 can communicate to version n (version n interfaces are supported)
- Preferably a standard on corporate level

Kittlaus, Fricker (2017): Software Product Management: The ISPMA-Compliant Study Guide and Handbook.

3. Product Planning
3.1 Product Requirements Engineering



3. Product Planning
3.2 Release Planning: Selection Criteria



Wohlin, Aurum: „What is Important when Deciding to Include a Software Requirement in a Project Release?“, 4th Intl Symp on Empirical Software Engineering. 2005.

3. Product Planning
3.2 Release Planning: Selection

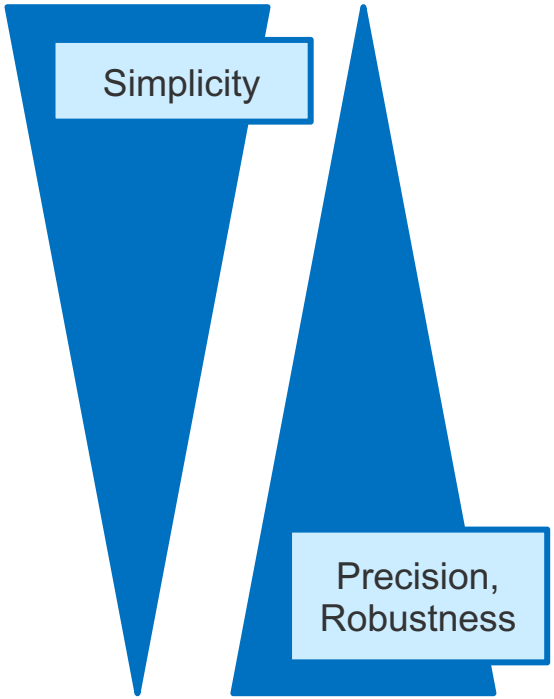
- Goal: to select those requirements, which maximize satisfaction of company objectives related to the software release
  - Fitness with product roadmap
  - Maximized value/cost ratio
  - Stakeholder satisfaction
  - Feasibility with respect to time and resources

- Need to select what to implement
  - Stakeholders (usually) ask for way too much
  - Balance time-to-market with amount of functionality
  - Decide which features go into the next release

- And what if stakeholders disagree?
  - Visualize differences in priority
  - Resolve disagreements

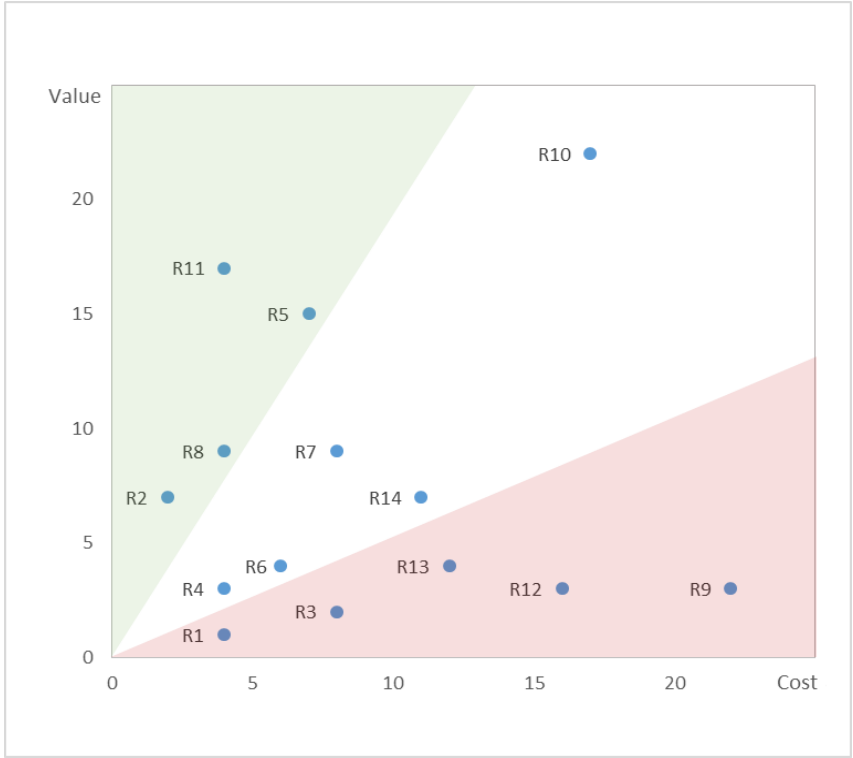
3. Product Planning
3.2 Release Planning: Prioritization Techniques

- Manual techniques
  - Cost value approach
  - Top-Ten
  - Numerical assignment (grouping)
  - Ranking (sorting)
  - Cummulative voting (100$ Test)
- Consensus-seeking techniques
  - Planning Poker
- Analytical techniques
  - Wiegers Prioritization Matrix
- Tool-based techniques
  - Integer Linear Programming (ILP with Expert Decisions Release Planner)
  - Analytical Hierarchy Process (AHP with IBM FocalPoint)

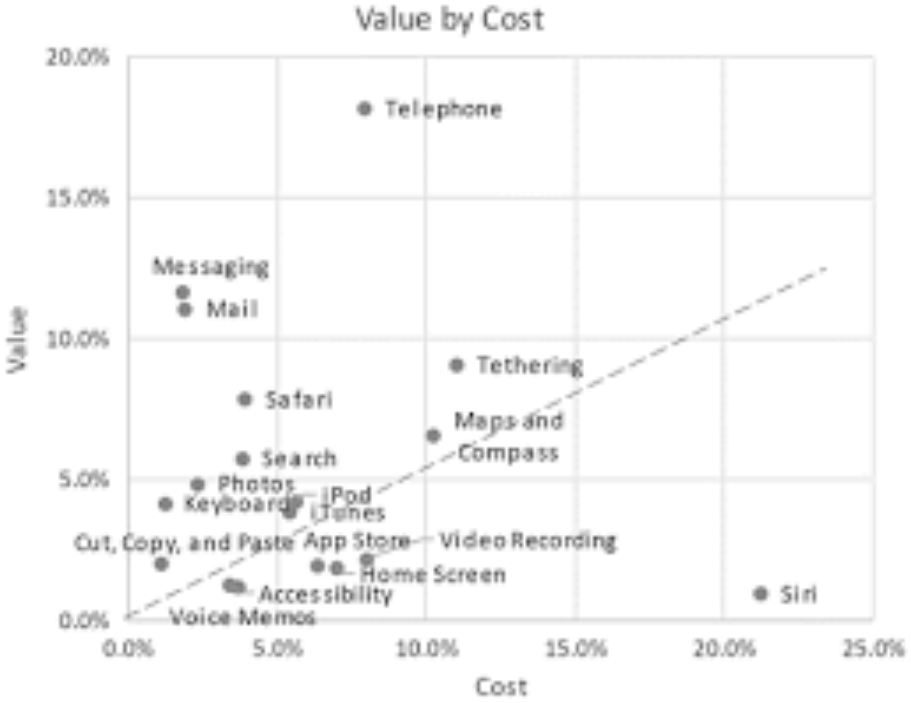


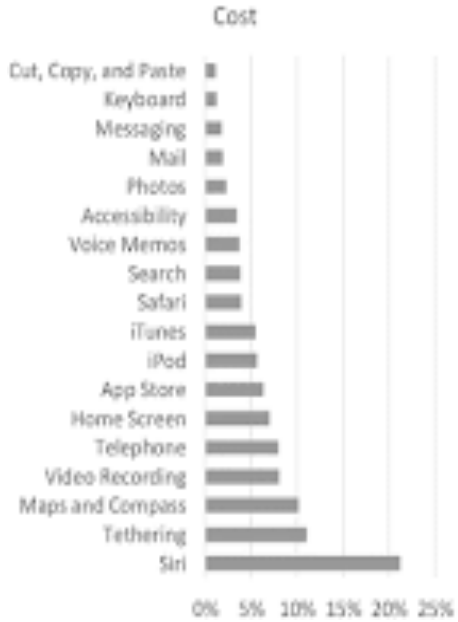
Berander, Andrews: „Requirements Prioritization“. In Arum, Wohlin: Engineering and Managing Software Requirements. Springer. 2005.

Cost Value Approach

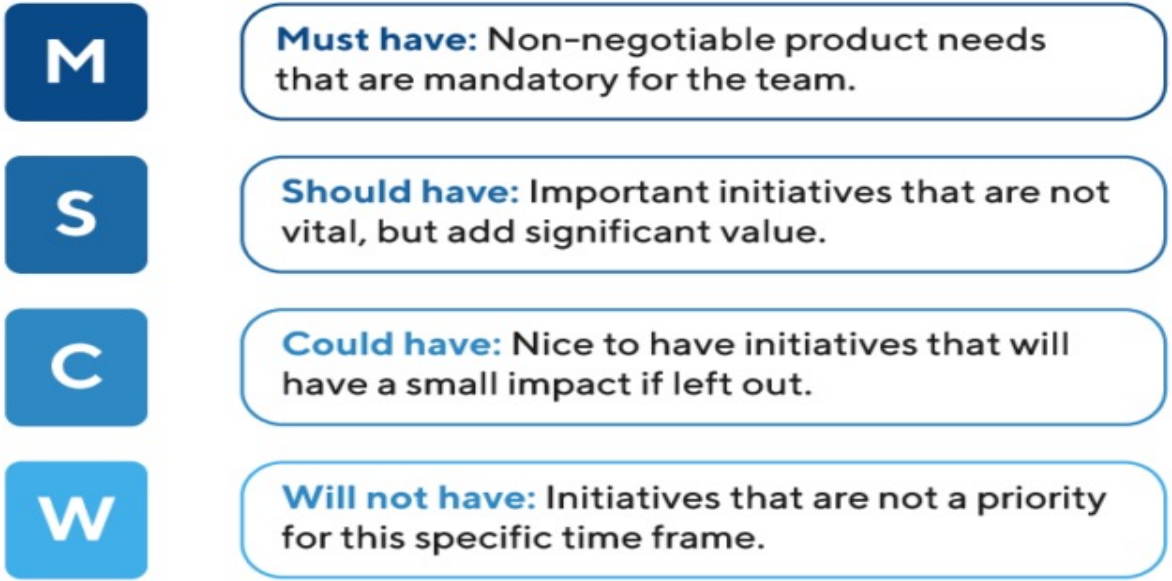


3. Product Planning
3.1 Product Requirements Engineering

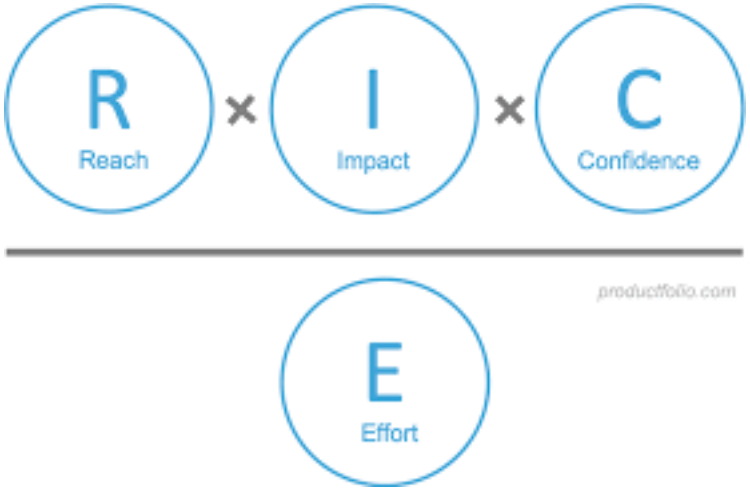




MoSCoW Prioritization



3. Product Planning
3.2 Release Planning: Prioritization Techniques



3. Product Planning
3.2 Release Planning: Prioritization Techniques

Cumulative Voting (or: 100$ Test)

- Each stakeholder distributes a total of 100 points (or $ or € or coins) on the requirements.
- The product manager sums up the points and presents the derived ordering of the requirements.

- Facebook example:

	Stakeholder 1	*Stakeholder 2*	*Stakeholder 3*	*Total*
Layout customization	10	20	5	35
Dislike button	30	20	25	75
Photo app integration	25	20	20	65
Profile visit stats	**25**	**30**	**35**	**90**
Email integration	10	10	15	35

Numerical Assignment (or: Priority Groups)

- Each stakeholder groups requirements into different priority groups (e.g. critical, important, useful).
- The product manager sums up the weights (e.g. critical = 9, important = 3, useful = 1).

- Facebook example:

	Stakeholder 1	*Stakeholder 2*	*Stakeholder 3*	*Total*
Layout customization	useful	important	useful	5
Dislike button	**critical**	**important**	**important**	**15**
Photo app integration	important	important	important	9
Profile visit stats	**important**	**important**	**critical**	**15**
Email integration	useful	useful	useful	3

Ranking (or: Sorting)

- Each stakeholder sorts requirements in decreasing order.
- Product manager sorts requirements by considering the average or median priority of each requirement.

Stakeholder 1	*Stakeholder 2*	*Stakeholder 3*
Req. 2	Req. 4	Req. 4
Req. 3	Req. 1	Req. 2
Req. 4	Req. 2	Req. 3
Req. 1	**Req. 3**	**Req. 5**
Req. 5	Req. 5	Req. 1

Rank	*Requirement*	*Average*
4	Email integration	3,67
2	Dislike button	2
3	Photo app integration	3
1	**Profile visit stats**	**1,67**
5	Layout customization	4,67

Top-10 Requirements

- Each stakeholder selects 10 favorite requirements.
- Product manager sorts requirements by considering fairness and satisfaction of stakeholders.

3. Product Planning
3.2 Release Planning: Prioritization Techniques

Planning Poker: Team Consensus for relative effort and value

3. Product Planning: 3.2 Release Planning
Release Plan

- Formal document (sometimes called release definition)
- At least a documented list of evaluated requirements that are to be implemented in the release
- Better a comprehensive release specification with references to individual requirements (as described on the following charts)
- Sign-off by all stakeholders required
- Level of detail dependent on development approach (waterfall, iterative, agile …)

3. Product Planning: 3.2 Release Planning Release Plan

- Include short descriptions and references to requirements
  - Not entire requirement specifications / conceptual solutions

- Include strategic content
  - Use release themes
  - Indicate release fit to overall product roadmap
  - Identify strategic direction

- Use consistent and sufficient detail

- Document sources of requirements
  - Content specifics
  - Source information: who and when

3. Product Planning: 3.2 Release Planning Release Plan

- To be written by Product Manager
  - Co-authors: Architect & Marketing
- Scope
  - Whole product release
  - Only for major and minor releases; not for bug fixes
- Content
  - Listing of product requirements to be incorporated in the next release
  - Dependencies between product requirements
  - Distributed ownership of work
  - Does not describe solutions, but refers to existing or planned Conceptual Solutions

3. Product Planning: 3.2 Release Planning
Release Plan Validation

- Roadmap fit
- Investments in resources
- Various ways:
  - Presentation for the company board (and other internal stakeholders)
  - Release business case
  - Return On Investment (ROI) Estimation

3. Product Planning: 3.2 Release Planning
Release Change Management

- What to do in case of
  - extra requirements forced by the company board
  - a delay due to an absent engineer
  - an opportunity from a customer or prospect

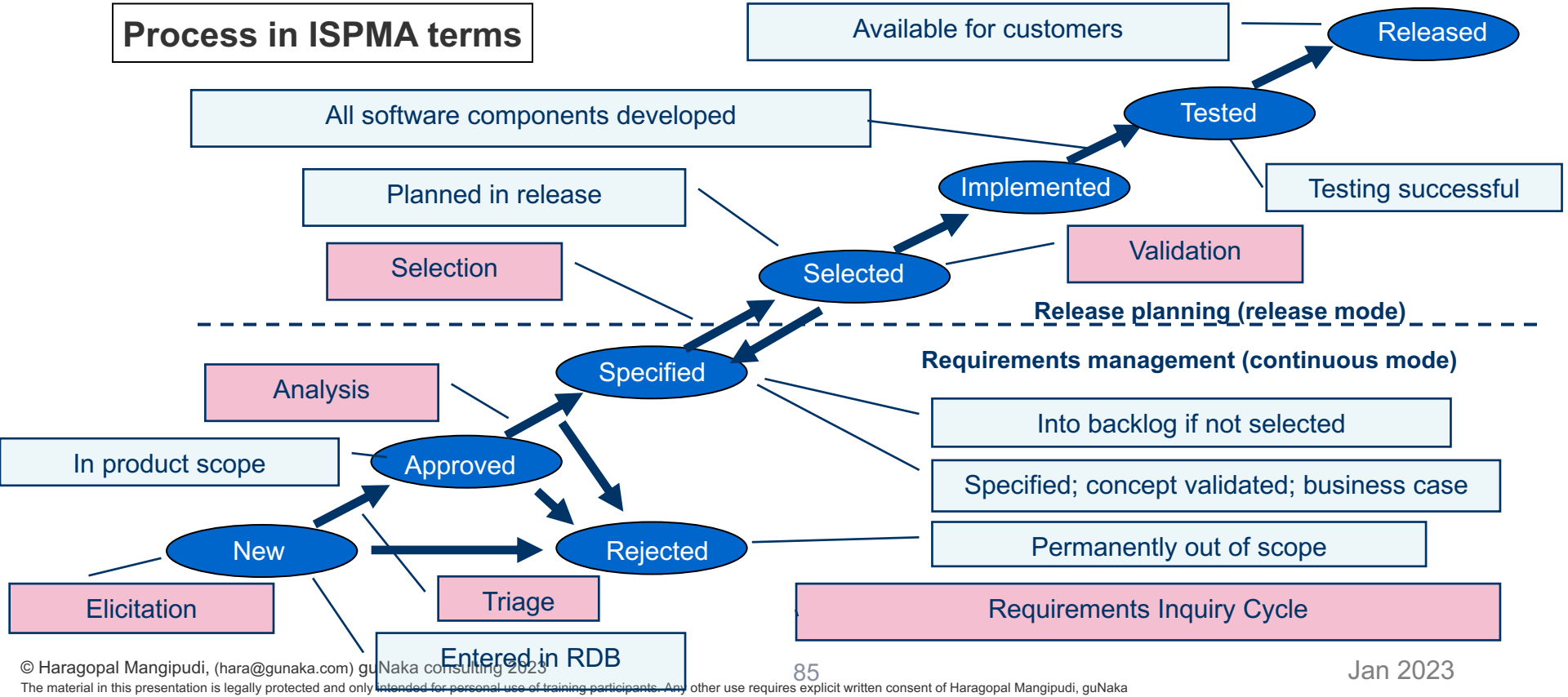
  during the development phase?

- Implementation of a scope change process (PRINCE2, ASL, SCRUM, etc.)

3. Product Planning: 3.2 Release Planning
Release Validation

- Ensure that the specified release is adequate, gets stakeholder acceptance, and meets its objectives
- Internal validation:
  - Testing
  - Simulations
- External validation
  - Beta testing
  - Pilot
- Certification (in special cases)

3. Product Planning
3.2 Release Planning



Agenda

3. Product Planning

3.1 Product Requirements Engineering

- Role of Requirements Engineering in Software Product Management
- Inquiry cycle with elicitation, analysis, and validation

3.2 Release Planning

- Release Planning Process and its conflicts / Structure of Release Plan

3.3 Roadmapping

- Product Roadmap and its elements
- Sources of input / Usage of Roadmaps

3.4 Product Life Cycle Management

- Phases of the Life Cycle
- Performance Management

3.5 Impact From Development Methodologies

3. Product Planning: 3.3 Roadmapping Roadmap

- Translates product strategy into series of releases on a time axis
- How is the product going to develop over the strategic timeframe of up to 5 years?
- What is a smart sequence of developing, releasing, and evolving the solution?

Kittlaus, Fricker (2017): Software Product Management: The ISPMA-Compliant Study Guide and Handbook.

3. Product Planning: 3.3 Roadmapping Roadmap

- Contents:
  - Timescale
  - Market and technology trends
  - New releases or versions and their (tentative) schedule
  - Release themes and technologies
  - Feature level as lowest level of detail
  - Target markets
  - Dependencies on other platforms, products, or technologies
  - Short term (e.g. 1-2 years): detailed and rather reliable.
  - Long-term (e.g. 5 years): less precise and subject to change
  - Assumptions
  - Legal disclaimer (for external audience)

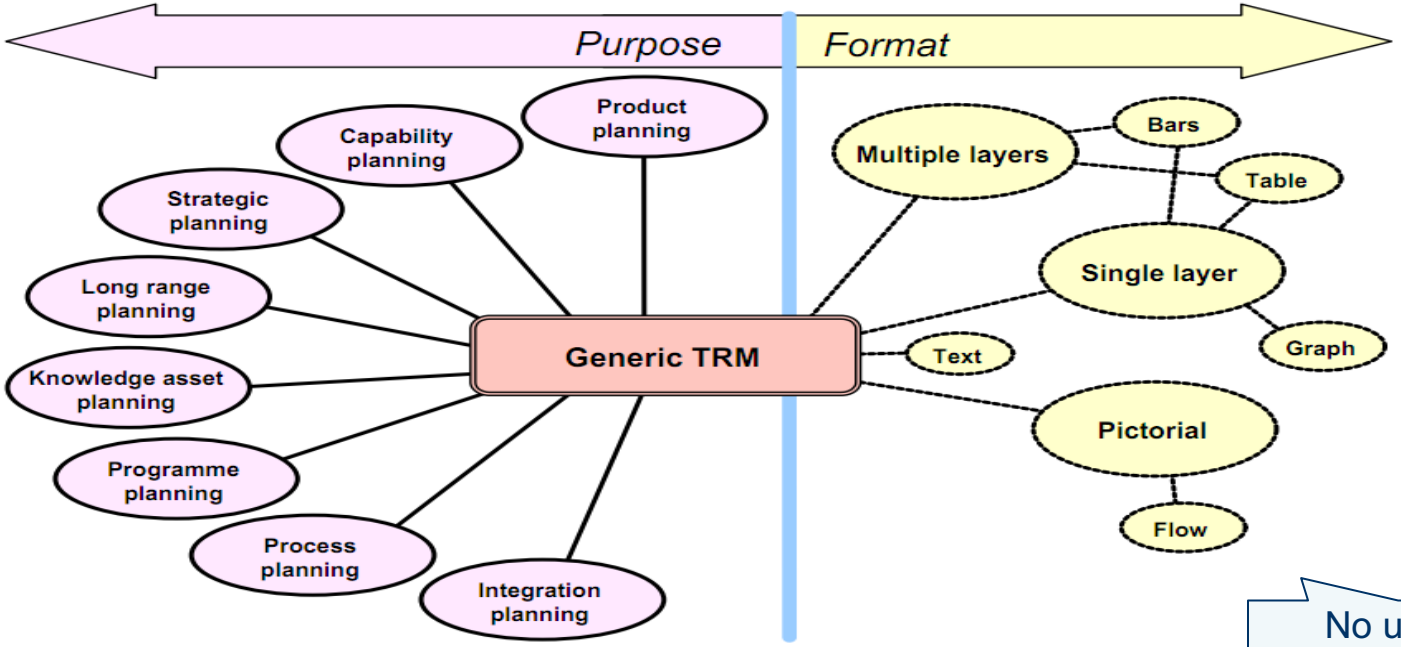
Kittlaus, Fricker (2017): Software Product Management: The ISPMA-Compliant Study Guide and Handbook.

3. Product Planning: 3.3 Roadmapping
Roadmap Purpose

- Give direction
- Communication to internal audience
  - Strategic alignment (iterative process for reaching long-term agreement on product direction and priorities)
  - Forecasting and Budgeting
  - Motivate developers, sales, or support to work on the product
- Communication to external audience (show subset)
  - Demonstration of product viability
  - Customers (Influence investment decisions)
  - Partners
  - Market research companies
  - Share holders / Venture capital funds (get funding)

Kittlaus, Fricker (2017): Software Product Management: The ISPMA-Compliant Study Guide and Handbook.

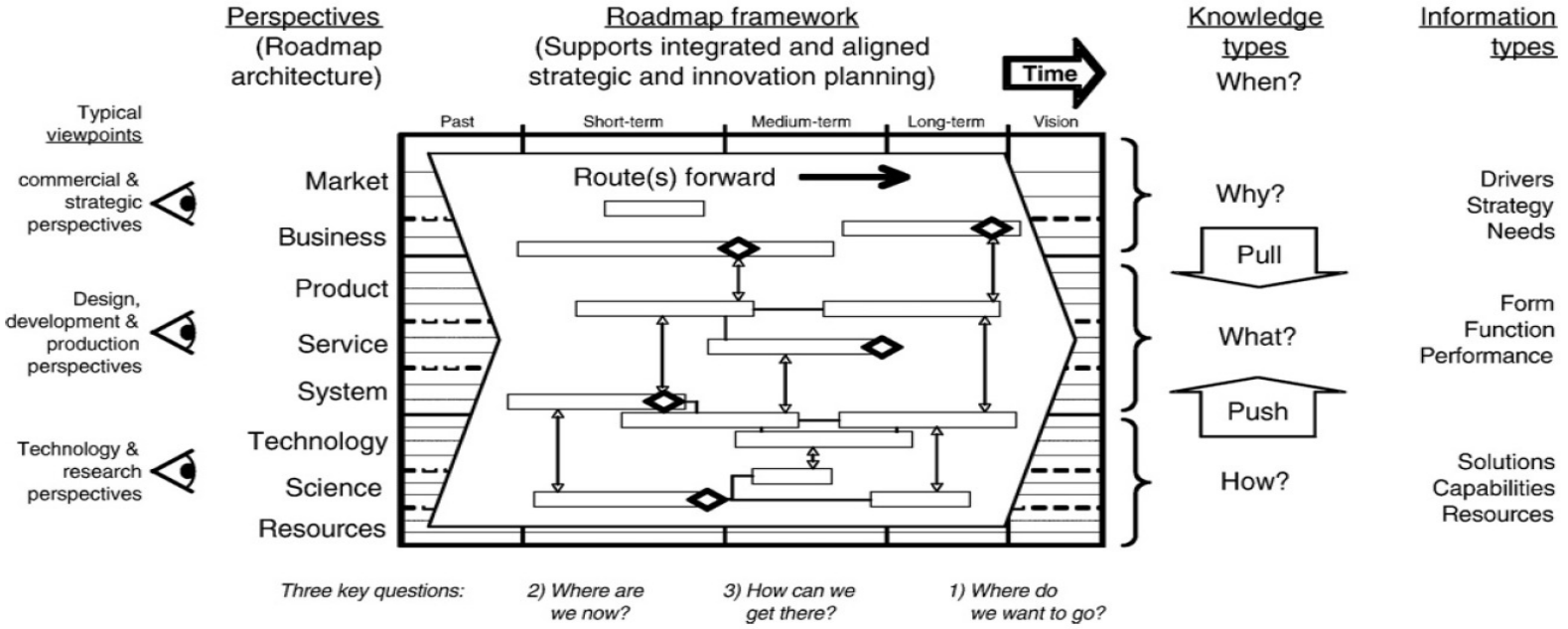
3. Product Planning
3.3 Roadmapping



No universal standard: chose and adapt to situation.

Phaal, Farrukh, Probert (2004): „Technology Roadmapping – A Planning Framework for Evolution and Revolution“, Technological Forecasting and Social Change.

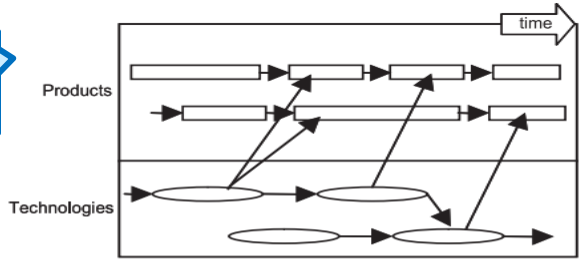
3. Product Planning
3.3 Roadmapping: Basic Structure

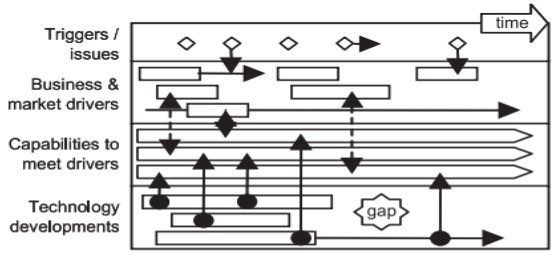


Phaal, Muller (2009): An architectural framework for roadmapping: Towards visual strategy. Technological Forecasting & Social Change 76:39-49.

3. Product Planning
3.3 Roadmapping: Examples

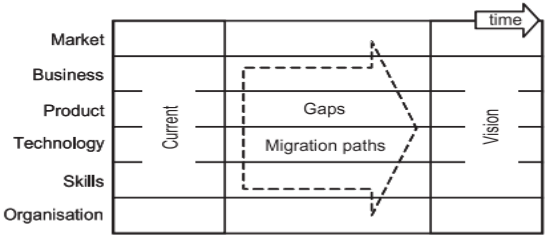
Product Planning

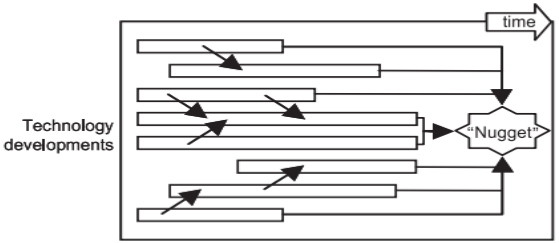




Service / Capability Planning

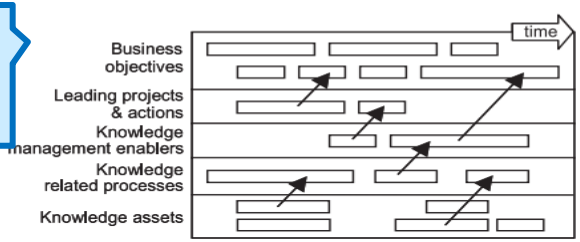
Strategic Planning

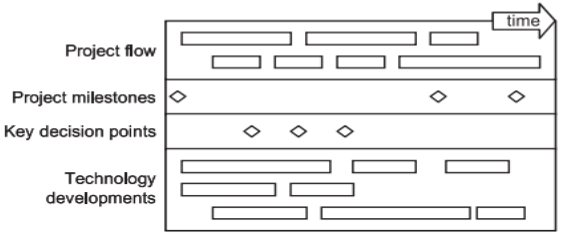




Long-Range Planning

Knowledge Asset Planning



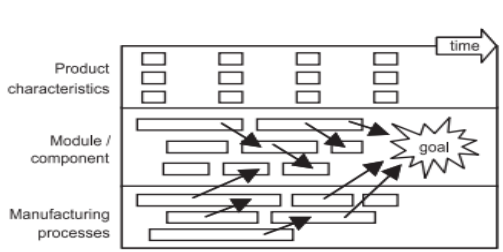


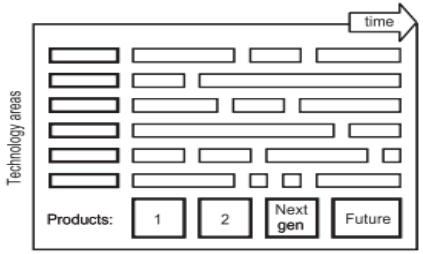
Program Planning

Phaal et al (2003): „Technology Roadmapping – Planning Framework for Evolution and Revolution". TFSC.

3. Product Planning
3.3 Roadmapping: Examples

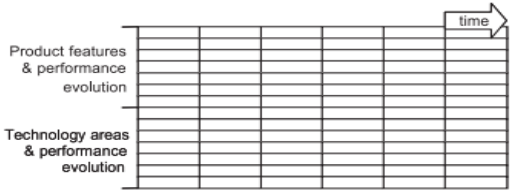
Layers

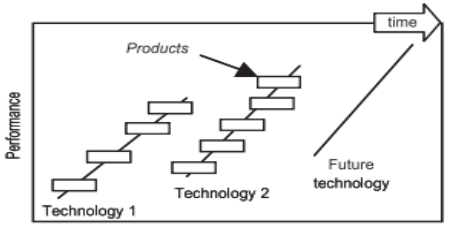




Bars

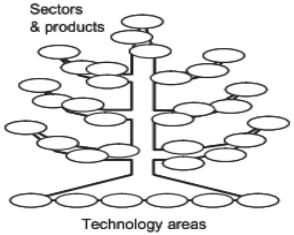
Tables

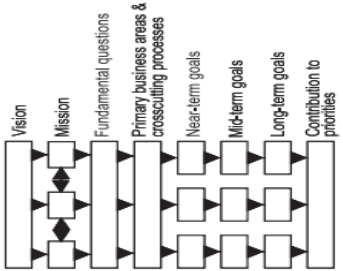




Graphs

Pictorial

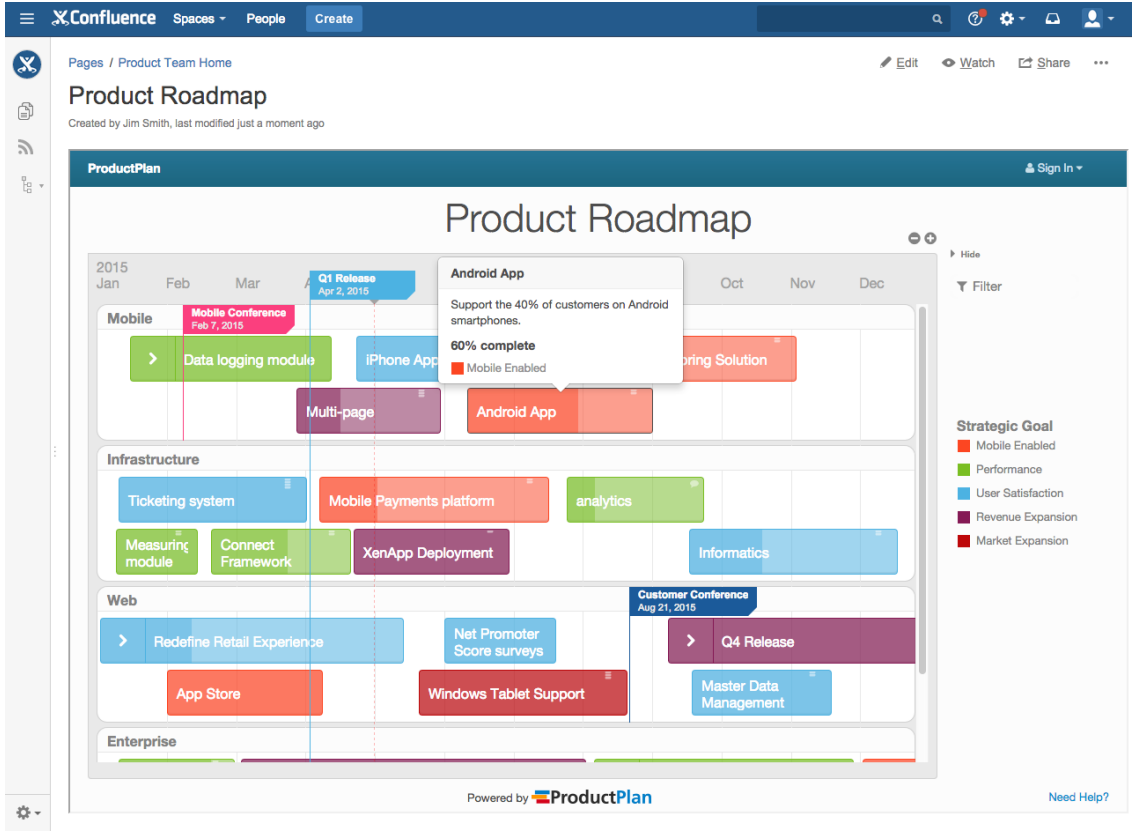




Flow Chart

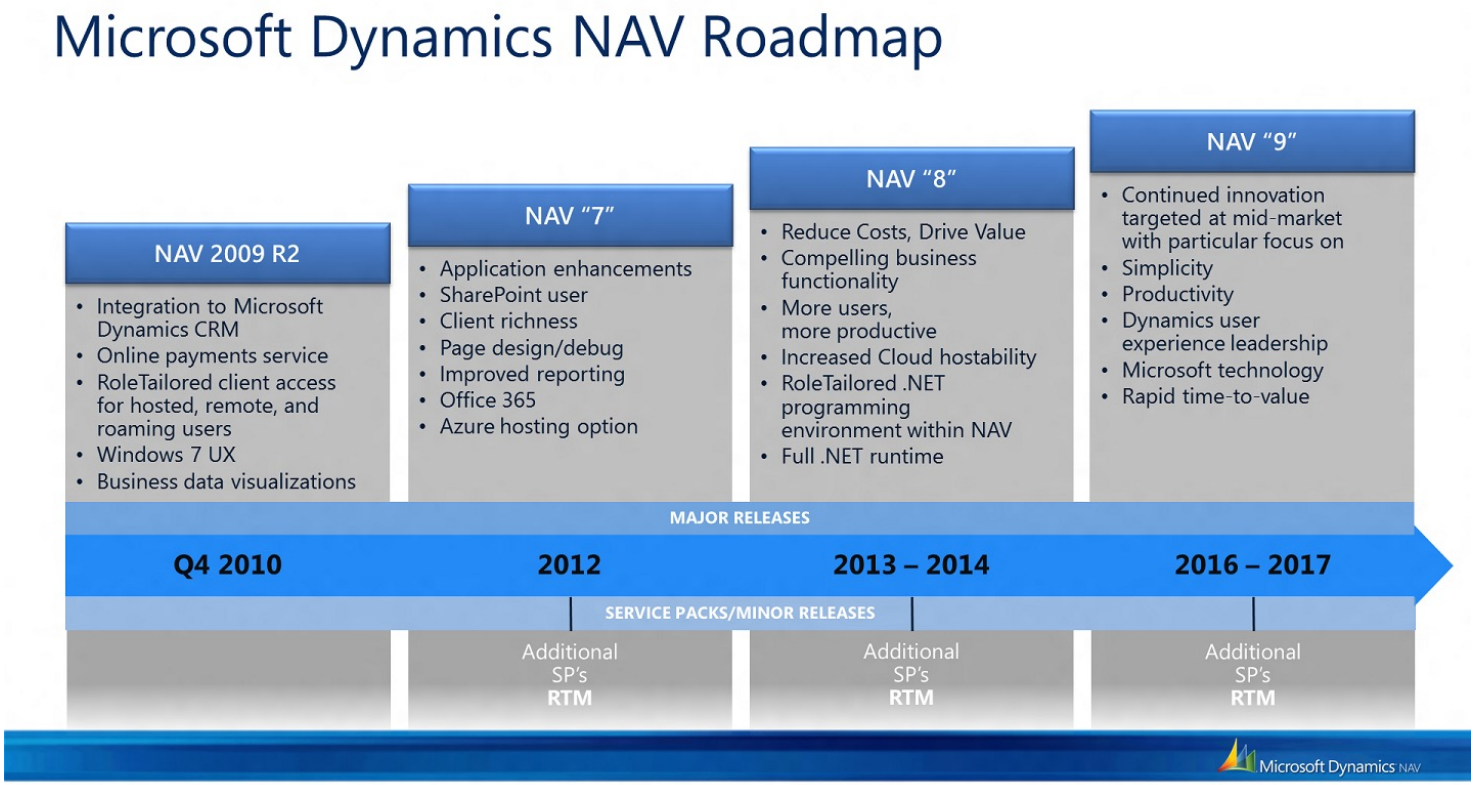
Phaal et al (2003): „Technology Roadmapping – Planning Framework for Evolution and Revolution“. TFSC.

3. Product Planning: 3.3 Roadmapping: Example Confluence

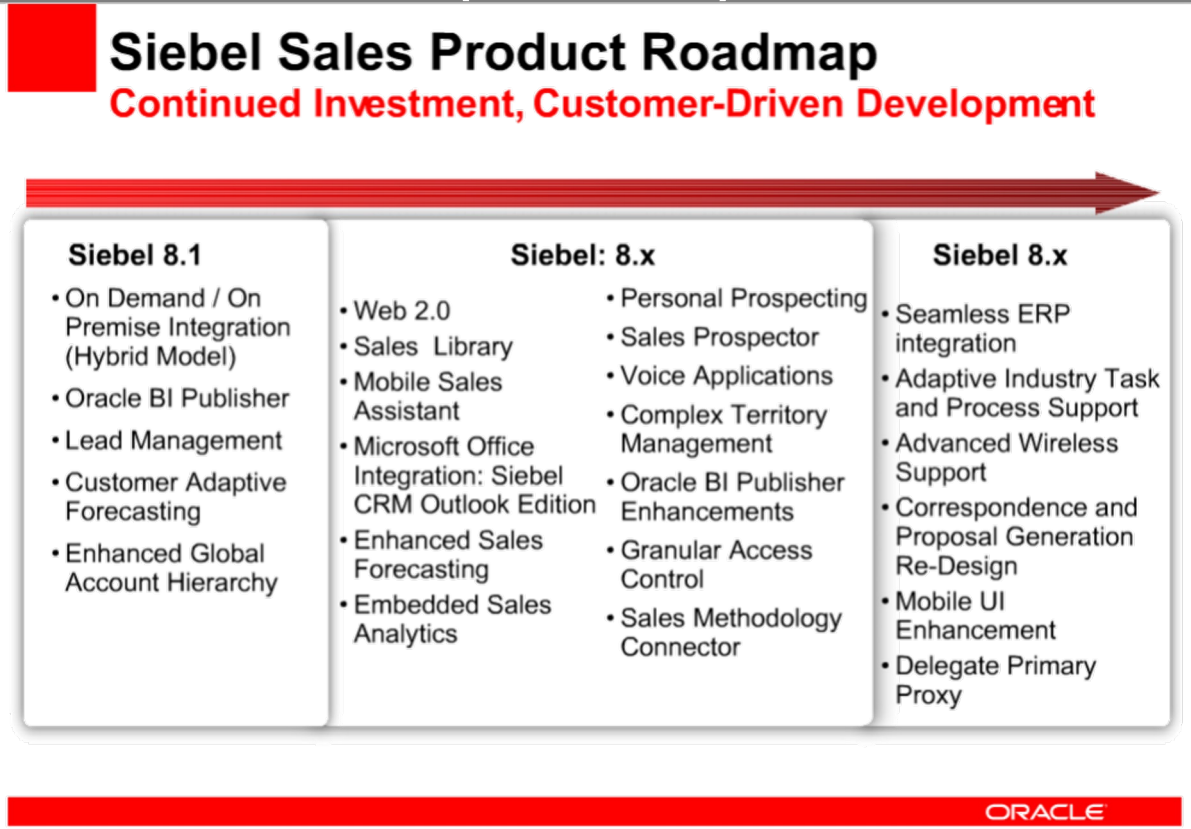


3. Product Planning: 3.3 Roadmapping: Example

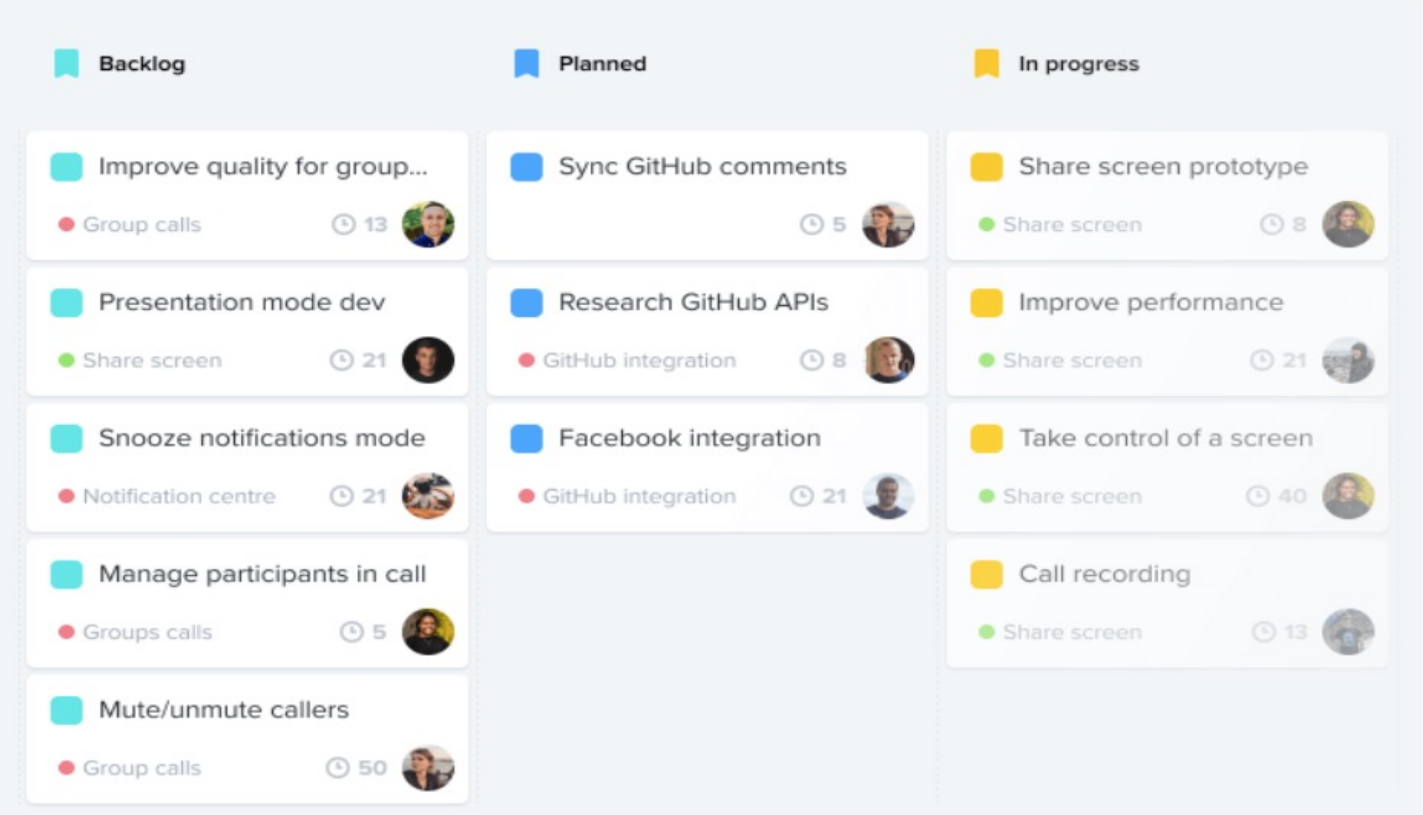
3. Product Planning: 3.3 Roadmapping: Example (external)



3. Product Planning: 3.3 Roadmapping: Example Oracle Siebel (external)



3. Product Planning: 3.3 Roadmapping: Kanban Sample



3. Product Planning: 3.3 Roadmapping: Quality Criteria

- Simplicity
  - Minimal representation of final outputs from strategy and planning.
  - Dialogue- and collaboration facilitating.
  - Documentation may be supplied to enhance a roadmap.
- Adaptation to purpose, situation, and culture (→ large variability)
  - Layers reasonably independent and consistent over time.
  - Effective for communication: influence decision-making.
- Correctness
  - Well-founded: based on company‘s best knowledge and expertise
  - Credible: representation of stakeholder interests and intentions
- Evolving
  - Iterative, exploratory development and adjustment.

Phaal et al (2003): „Technology Roadmapping – Planning Framework for Evolution and Revolution“. TFSC.
Phaal, Farrukh, et al (2003): „Customizing the Technology Roadmapping Approach“, Mgmt of Eng. and Techn.

3. Product Planning: 3.3 Roadmapping: Meta Process

- Form a roadmap team with key stakeholders (cross-functional, influencers), and assign the SPM as clearly designated owner of the product roadmap
- Create a 'roadmap knowledge base'
  - Roadmap templates
- Review and update product roadmaps regularly
- Compare your resources and key capabilities with your roadmap periodically

3. Product Planning: 3.3 Roadmapping: Process for Creation / Update

1. Which parties/persons should be involved?
   ➔ Update the roadmap team
2. Ensure up-to-date input
   - Corporate strategy and portfolio management decisions
   - Product strategy
   - Budgets and resources
   - Data from market and product analysis
   - Current development status compared to plan
   - Customer commitments

3. Product Planning: 3.3 Roadmapping: Process for Creation / Update

When you do it for the first time:

3. Choose a granularity level
    - Themes, components, product requirements, features
4. Choose which views you want to include in your roadmap
5. Choose a time scale

Everytime:

6. Go into iterative process to align roadmap with corporate strategy, product strategy, release plans, product RE

3. Product Planning: 3.3 Roadmapping: External Communication

1. Manage expectations
   - ‘Plan of intent’
   - Limit time horizon
   - No specific dates or details

2. Beware of the risk of information leaks
   - Assume competitors will see it
   - Use an NDA
   - Don’t leave it behind

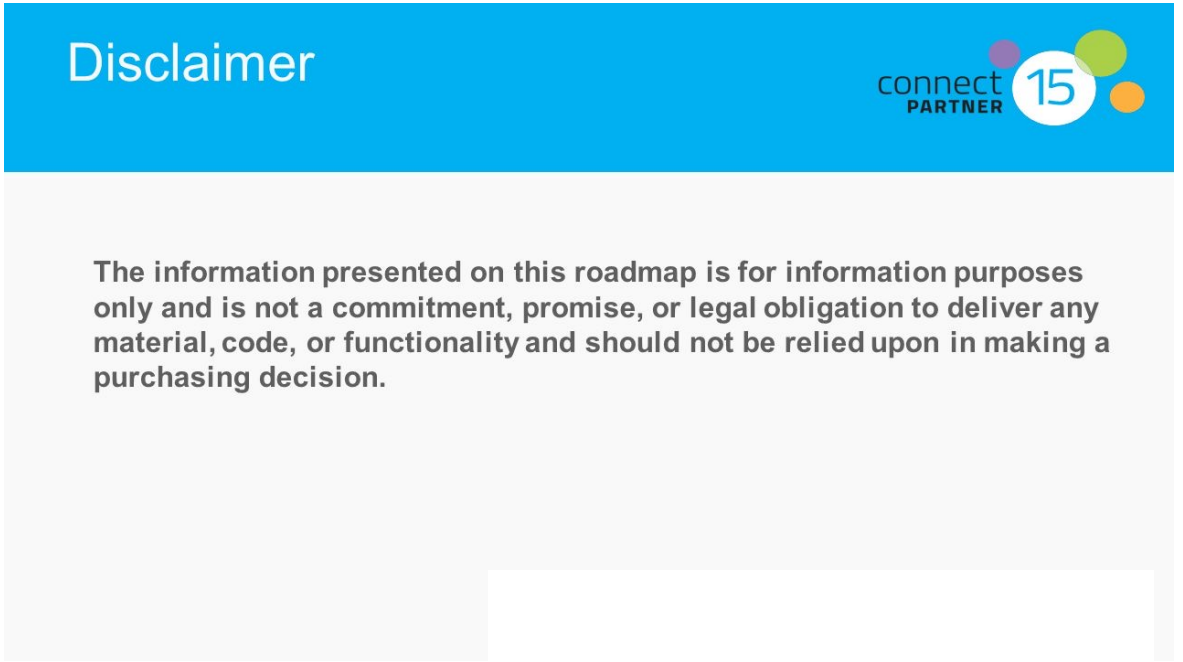
3. Avoid the delay of sales to future products
   - Present during a vision presentation
   - Offer “bridge” programs

4. Prepare for the “so what?” response
   - Know your market and customers
   - Use as a collaboration tool, not as a plan

3. Product Planning: 3.3 Roadmapping: Legal Aspects for External Audience

- Legal disclaimer for protection: Content is confidential, plans may change
- Non-disclosure agreement (NDA)

➔ Marketing value



Agenda

3. Product Planning

3.1 Product Requirements Engineering

- Role of Requirements Engineering in Software Product Management
- Inquiry cycle with elicitation, analysis, and validation

3.2 Release Planning

- Release Planning Process and its conflicts / Structure of Release Plan

3.3 Roadmapping

- Product Roadmap and its elements
- Sources of input / Usage of Roadmaps

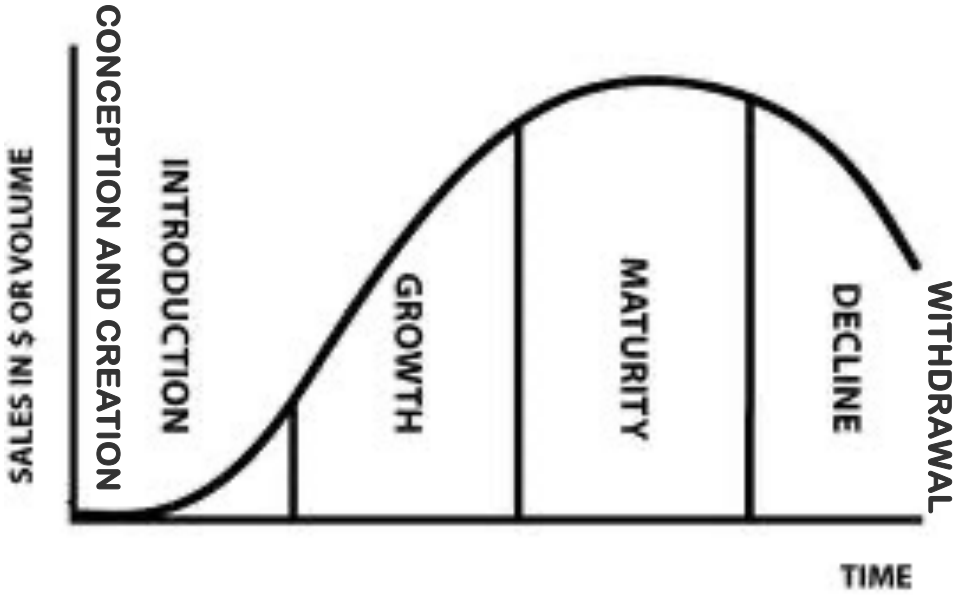
3.4 Product Life Cycle Management

- Phases of the Life Cycle
- Performance Management

3.5 Impact From Development Methodologies

3. Product Planning
3.4 Product Life Cycle Management

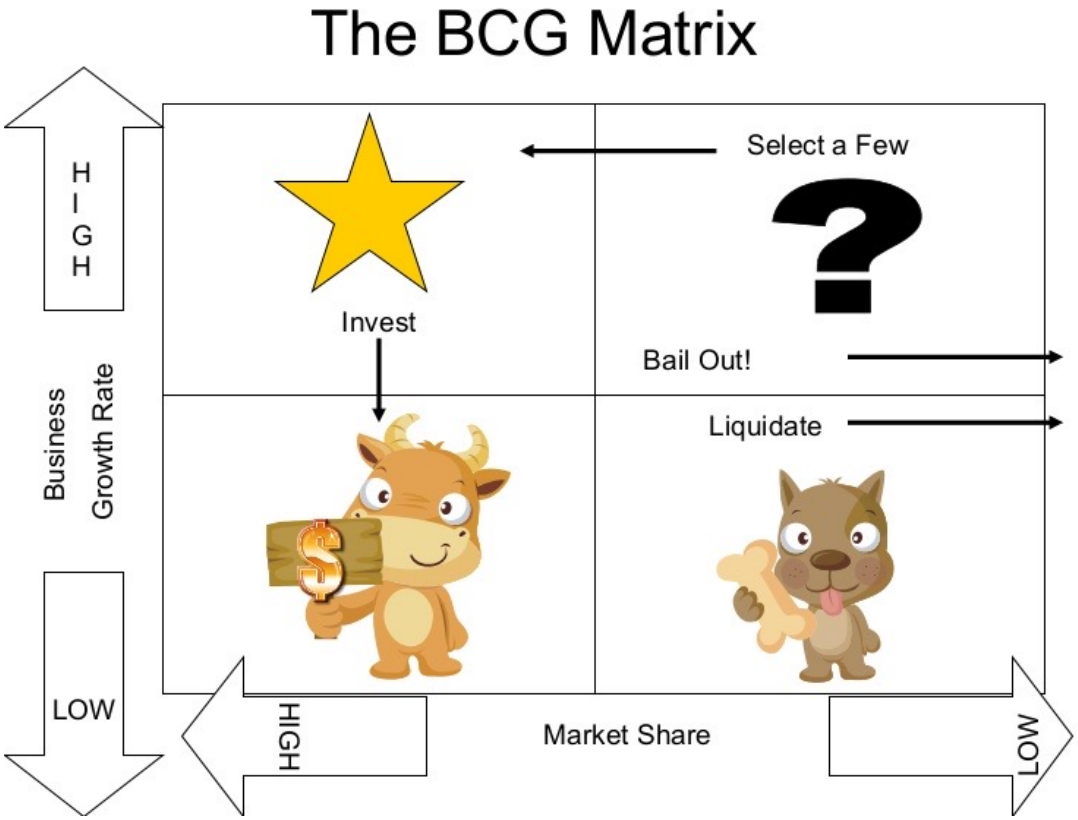
- Holistic approach from cradle to grave
- PLM approach (manufacturing industry) not really transferable to software



Software product across life cycle (all versions)

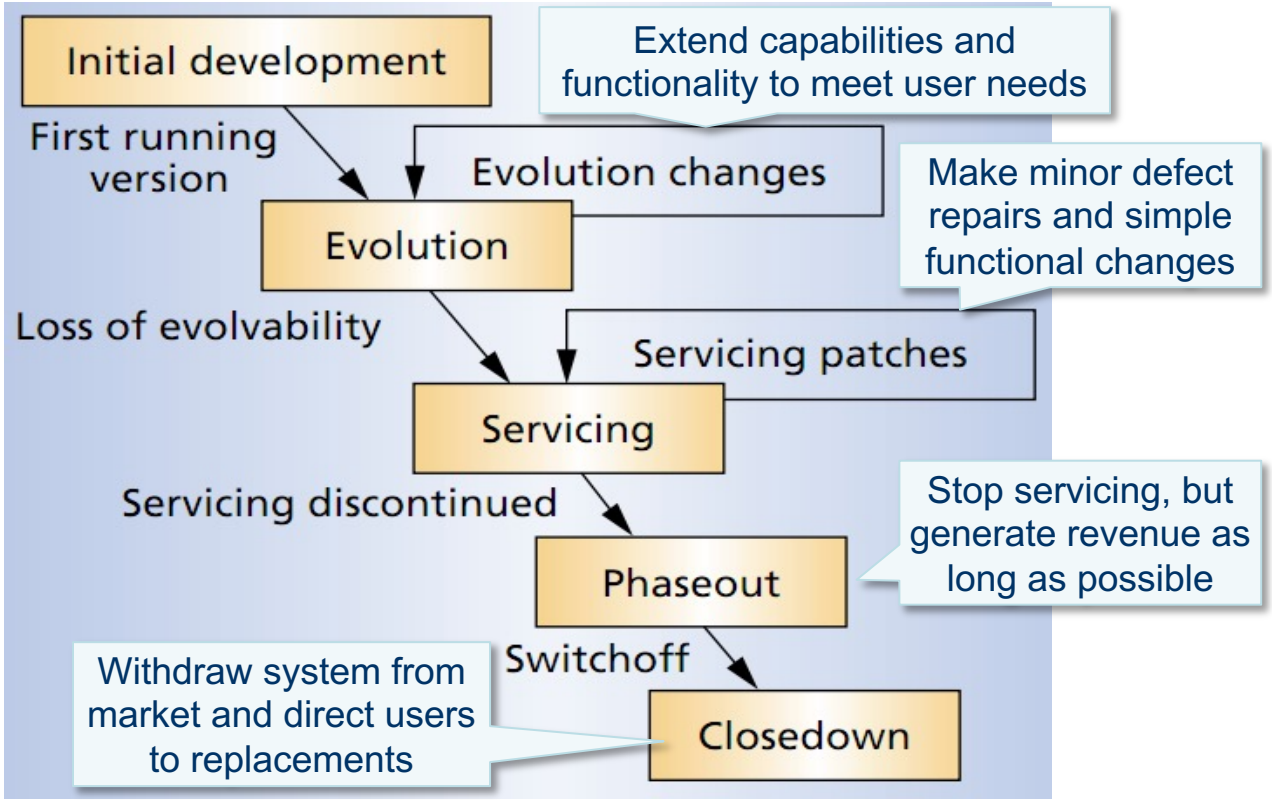
Phase	Focus Areas	Leading Stakeholders
Conception and creation	Innovation, positioning, investment	Research, Development, Marketing, regulatory bodies
Market introduction	Launch, market share, investment	Marketing, Sales
Growth	Market share, functionality, investment	Research, Development, Marketing, Sales
Maturity	Cash cow, revitalization, services	Sales, Services, Support
Decline	Customer retention, cash cow	Marketing, Sales, Support
Withdrawal	Customer retention, cost reduction	Marketing

3. Product Planning
3.4 Product Life Cycle Management



3. Product Planning
3.4 Product Life Cycle Management

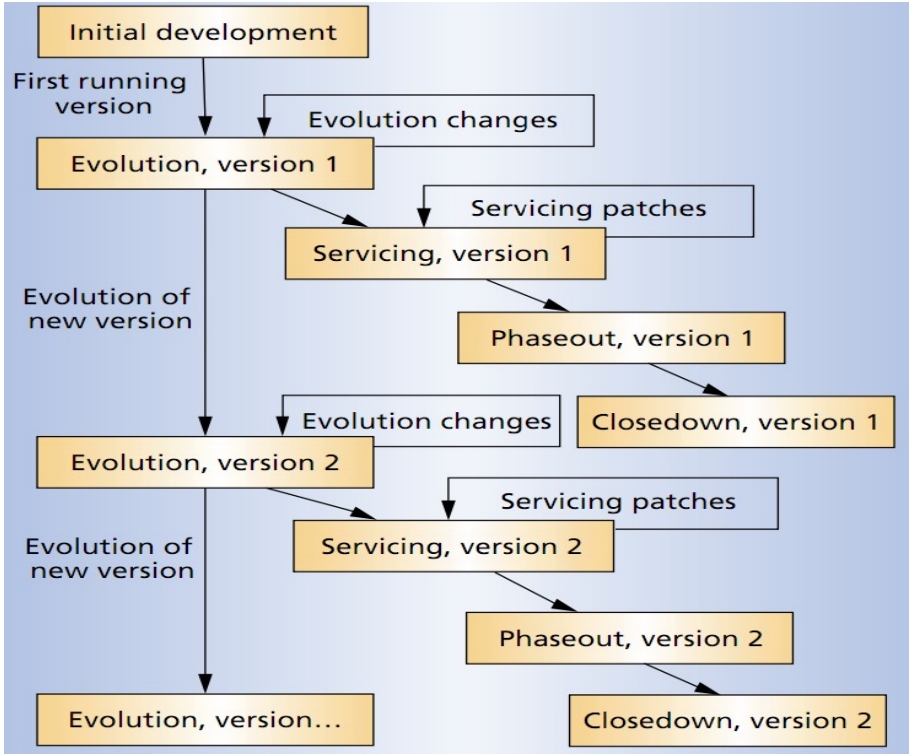
Software Product: one version



Rajlich, Bennett (2000): A Staged Model for the Software Lifecycle. IEEE Computer.

3. Product Planning
3.4 Product Life Cycle Management

Software Product: multiple versions



Rajlich, Bennett (2000): A Staged Model for the Software Lifecycle. IEEE Computer.

Withdrawal from Service

- No more maintenance
- Customers are expected to migrate
- Vendor can try to generate revenue from
  - customer-specific support
  - license fees (depends on terms and conditions)

Closedown

- Vendor shuts down the software
- Vendor directs users to a replacement system
- Residual responsibilities
  - Source code retention
  - Legal liability
- Migrate data to a new system

Rajlich, Bennett (2000): A Staged Model for the Software Lifecycle. IEEE Computer.

Knowledge Management

- Ensure that the knowledge required for the viability of the product continues to be accessible and available for the company during the product life cycle
- Special problem for legacy products
  - incentives can help

Performance Management

- Continuous measurement required to drive timely corrective actions
  - Product profitability
  - Actual vs. planned revenue
    - New vs. existing customers
    - Product sales vs. maintenance
  - Customer satisfaction
  - Market share

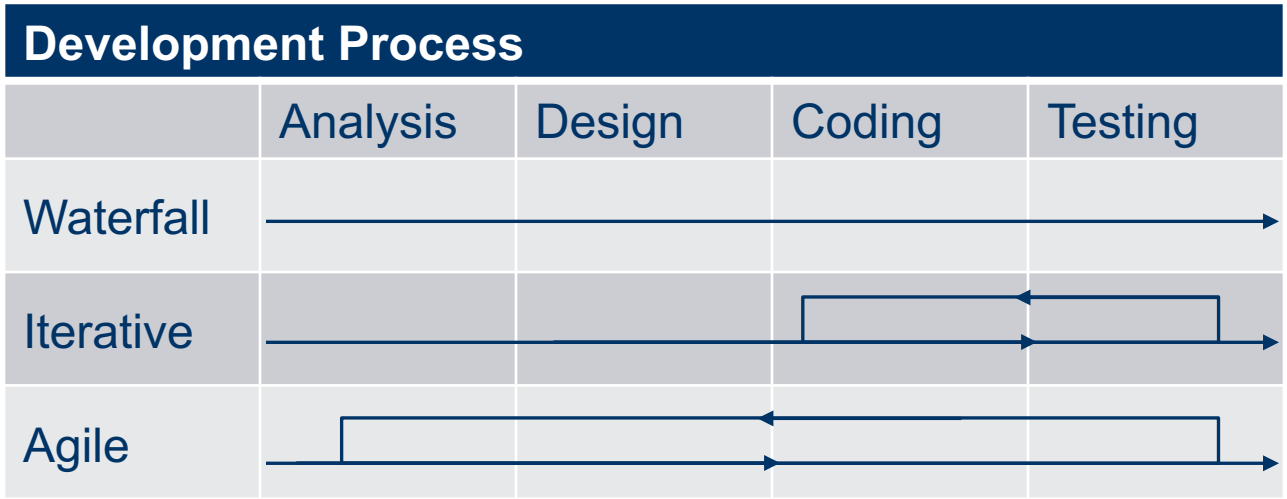
3. Product Planning: 3.1 Product Planning Approaches Impact from Development Methodologies

What is a development methodology?

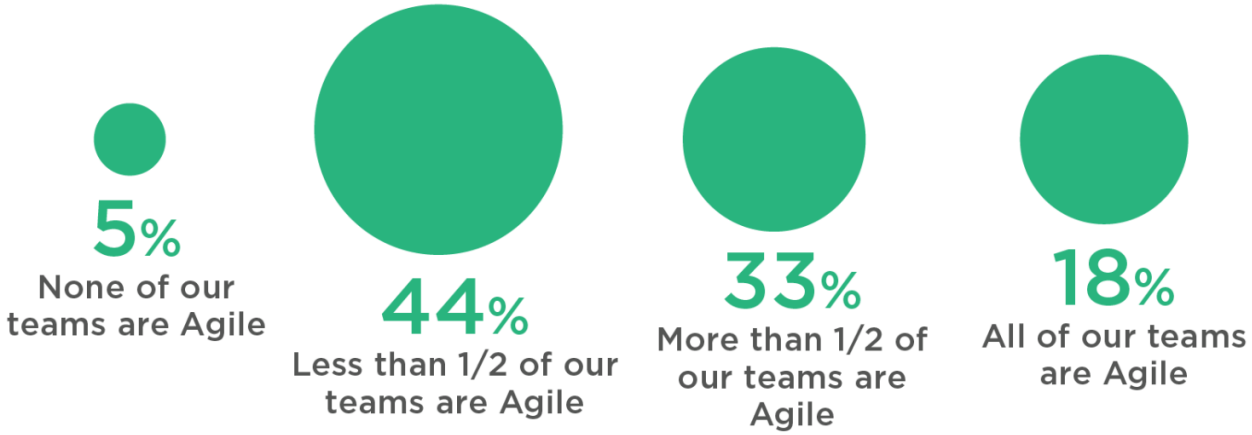
- A framework that is used to structure, plan, and control the process of developing a software system
- Roles, activities and artifacts for developing software

FreeDigitalPhotos.net Vichaya Kiatying-Angsulee

3. Product Planning: 3.1 Product Planning Approaches Impact from Development Methodologies



3. Product Planning: 3.1 Product Planning Approaches Impact from Development Methodologies

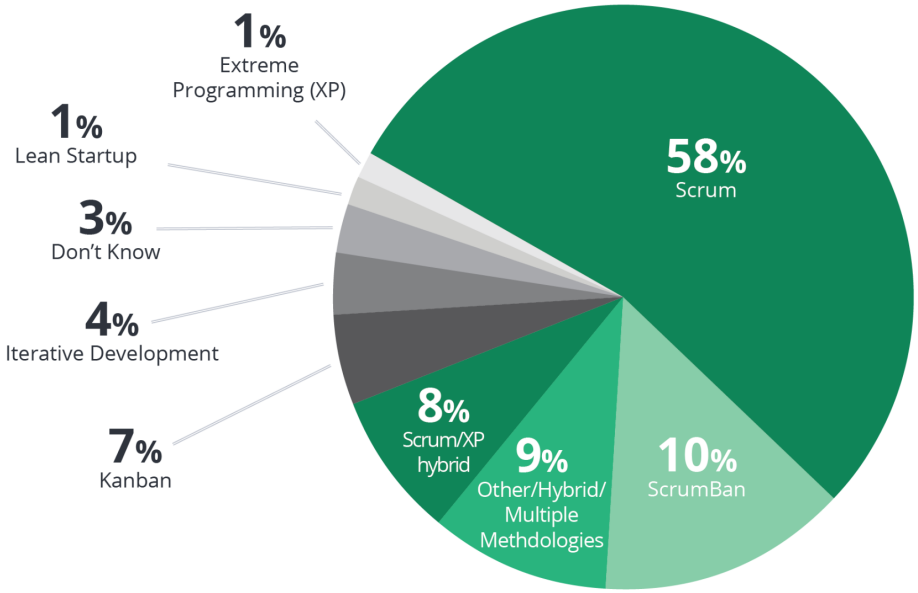


Percentage of Teams Using Agile

Survey: more than 1100 participants worldwide (for 2020)

CollabNet VersionOne: 14th Annual State of Agile Survey, 2020

3. Product Planning: 3.1 Product Planning Approaches Impact from Development Methodologies



CollabNet VersionOne: 14th Annual State of Agile Survey, 2020

3. Product Planning: 3.1 Product Planning Approaches Impact from Development Methodologies

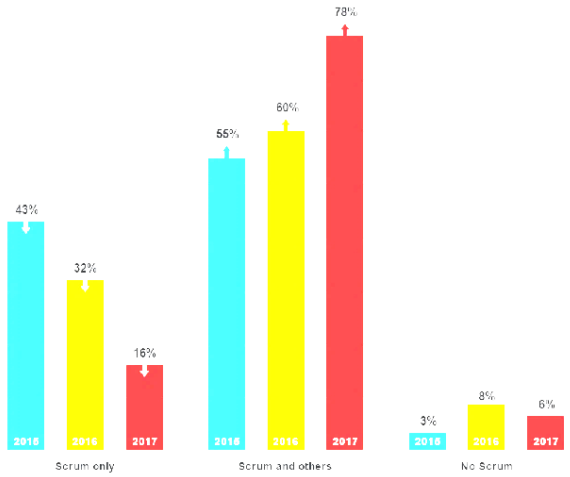
What agile approach is your organization using?

AGILE APPROACHES

Survey:

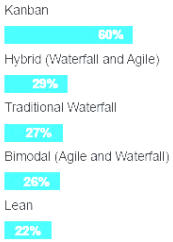
More than 2000 participants

from 91 countries



Other frameworks used

The average respondent reported using 3.5 frameworks



Scrum Alliance State-of-Scrum Report 2017/18

3. Product Planning: 3.1 Product Planning Approaches Impact from Development Methodologies

Scrum Roles

Role	Description
Product owner	As advocate for customers and users, determines what work will be done via the backlog. Accountable for product.
Scrum master	Helps team follow the process and removed impediments. Accountable for process.
Team	Group of full-time professionals that develop the product. Accountable for development.

FreeDigitalPhotos.net

Scrum Ceremonies

Activity	Description
Sprint planning	User stories are accepted from backlog by team
Daily Meeting	Team shares individual status: • What have I done since last meeting • What will I do by next meeting • What are my impediments
Sprint review	Team presents work related to accepted user stories
Sprint retrospective	Each team member shares their impressions/learnings from the sprint by stating what should team: • Start doing • Stop doing • Continue doing

Scrum: Different definitions of terms

- Vision: more detailed (feature level), shorter term (replacement for high-level functional specification)
- Product: not defined, sometimes meant as result of a development project
- Roadmap: more like a development plan than a strategic document
- Release: deliverable given to a customer; sometimes meant as result of a sprint, i.e. a phase of the development project
- Product Owner: role in a Scrum Team, not meant as the head of a business unit
- Project: not defined, but frequently used
- Backlog: prioritized list of requirements

➔ Make sure there is a common understanding of the terms used in your company

3. Product Planning: 3.1 Product Planning Approaches Impact from Development Methodologies

SPM vs. Product Owner

- Overlapping responsibilities
- Different time horizons
- In small organizations: SPM assumes Product Owner role
- In larger organizations:
  - SPM cannot do it all

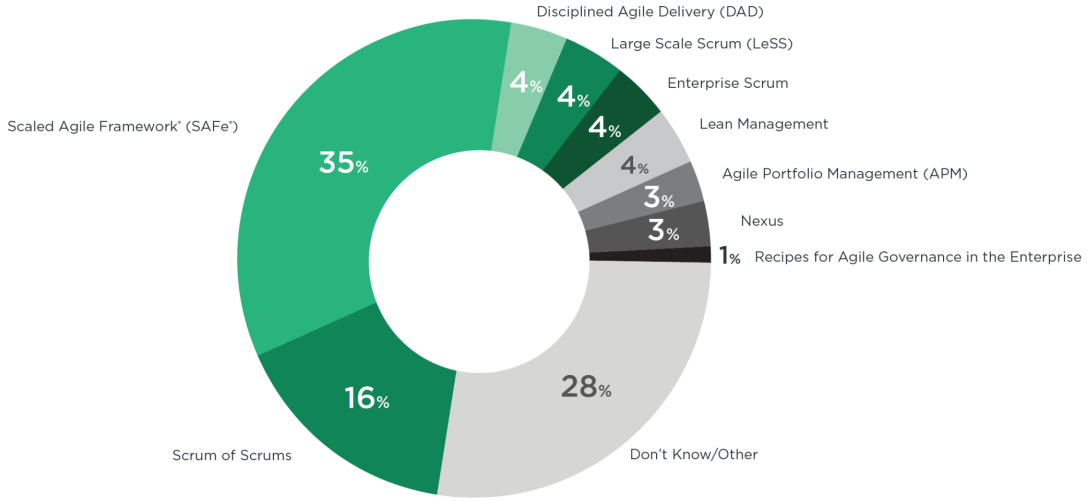
  ➔ Keep it separate
  - Product Owner is part of the Development organization with a strong dotted line into SPM or
  - Product Owner is part of the SPM organization and delegated into the development team

Leffingwell (2011): Agile Software Requirements, Addison Wesley
Kittlaus (2012): Software Product Management and Agile Software Development: Conflicts and Solutions , In: Maedche, A., Botzenhardt, A., Neer, L. (Ed.): Software for People - Fundamentals, Trends and Best Practices, Springer

3. Product Planning: 3.1 Product Planning Approaches Impact from Development Methodologies

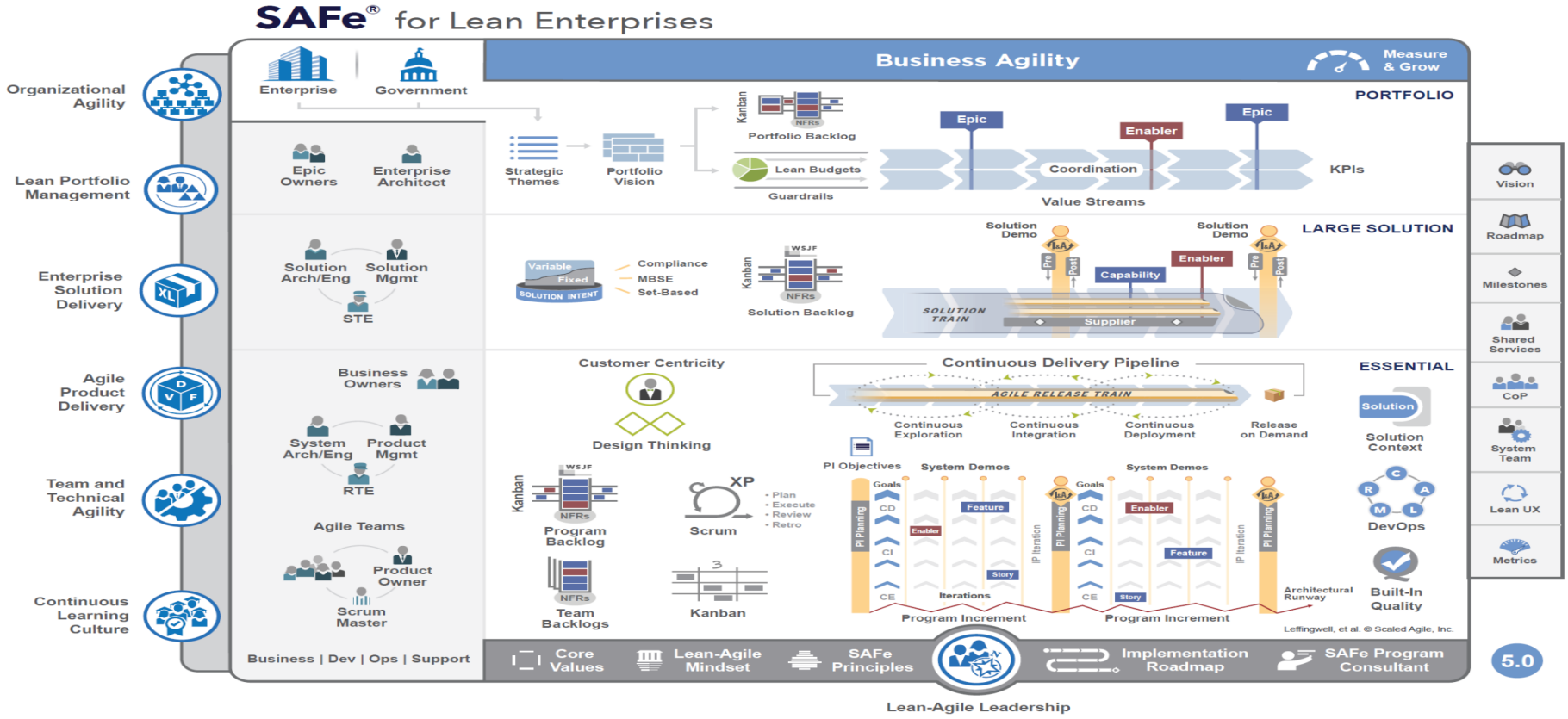
Scaling Methods and Approaches

The Scaled Agile Framework continues to be the most popular scaling method cited by respondents.



CollabNet VersionOne: 14th Annual State of Agile Survey, 2020

3. Product Planning: 3.1 Product Planning Approaches Impact from Development Methodologies



Legal Statement