

Bespoke and Market Driven RE, Tools

2012-10-16

Emil Alégroth

Who am I writing to?

- Identify the stakeholders
 - Identify where in the process you are
 - Identify a suitable writing level
 - For CEO?
 - Product managers?
 - Developers?
 - The customer's customers?
 - Etc.
 - What is the document going to be used for?
 - Proposal?
 - Design?
 - Detailed specification?
 - Etc.
-

Who/Why model

Business
Centered



Technical
Detail

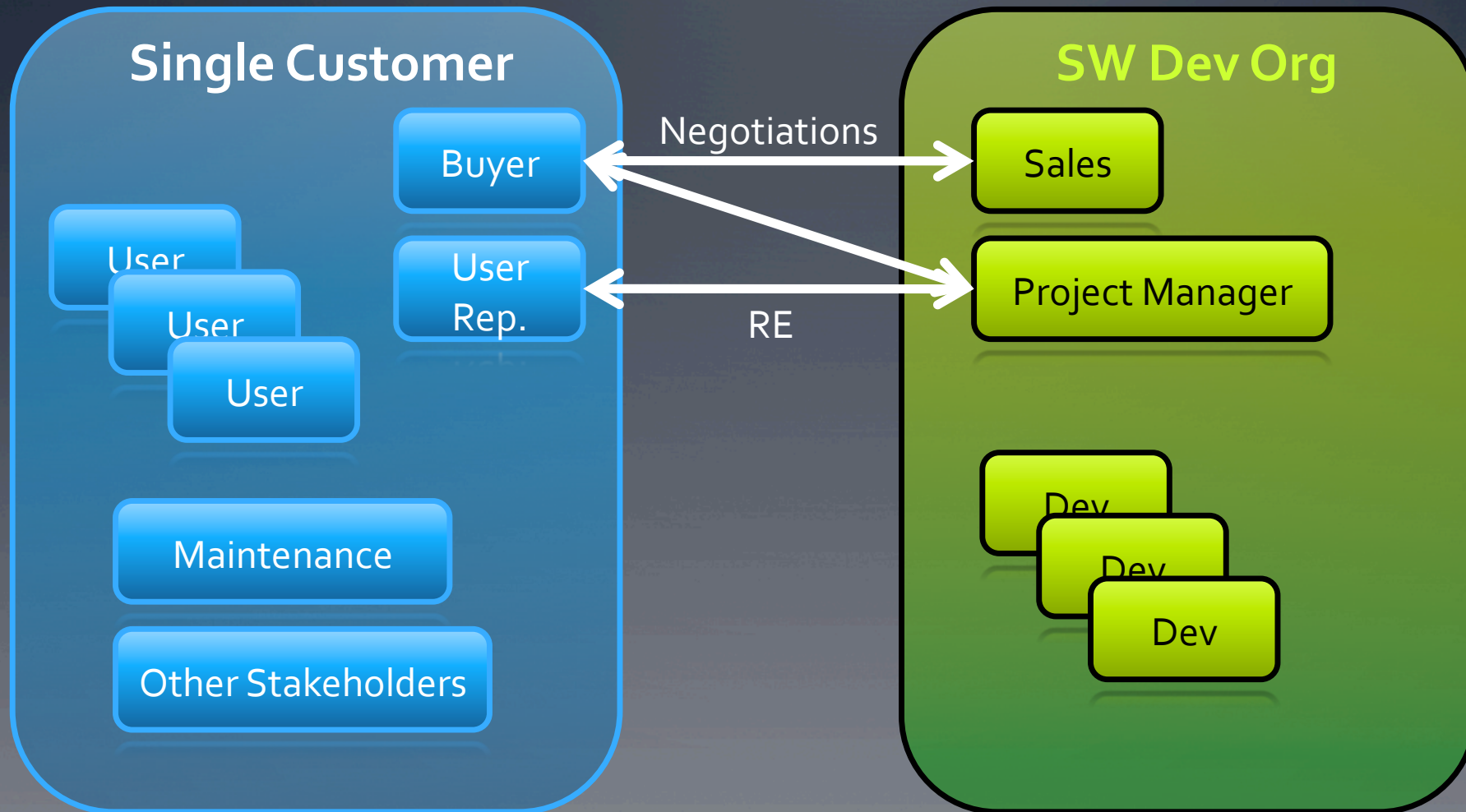
High-
Level



Low-
Level

Business level / Stakeholder	CEO	Product manager	Project manager	Developer/Tester
Interview notes	Blue box			
Product proposal				
Product spec.				
Project spec.			Blue box	
SRS				
Detailed SRS				Blue box
Tasks				

Bespoke Software Development



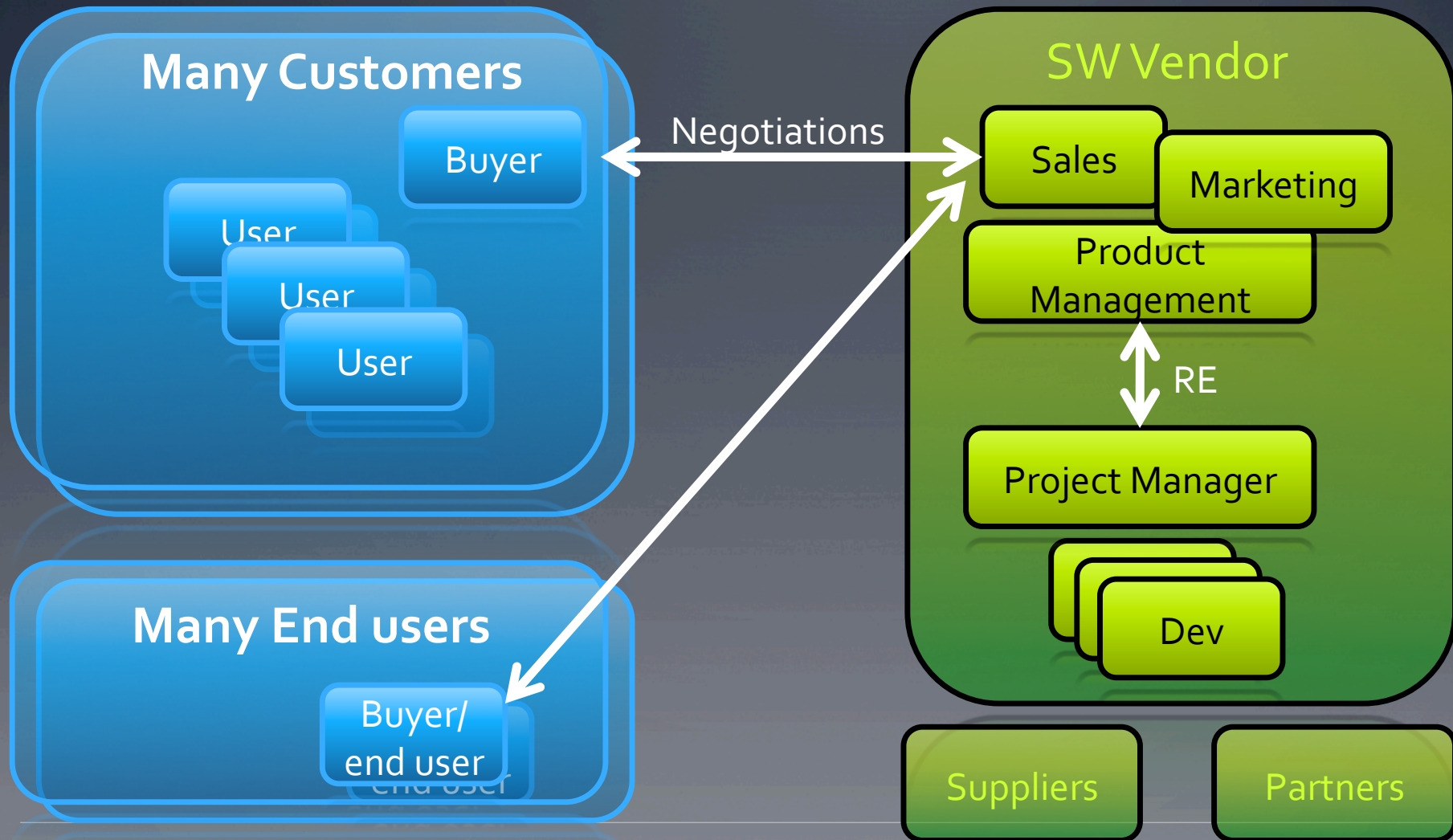
Bespoke Software Development

- **Also known as:** Custom/Traditional Software Development
 - RE primarily startup activity
 - Pre-study/Feasibility study, Contract
 - SW Req Specification (SRS)
 - Changes require negotiations
 - **Project focus** (RE, Analysis, Design, Impl, V&V, Release)
 - Domain knowledge from customers/users
 - Success = contractual fulfillment & customer satisfaction
-

Bespoke Software Development – RE steps

1. Customer states need in general terms in **Request for Proposals (RFP)**
 2. **Dev company creates proposal** = approach, prelim requirements, schedule, budget
 3. **Customer selects best proposal**
 4. **Dev company prepares SRS & presents**
 5. **Changes** => prioritization & negotiations
 6. **Budget/Schedule problems** => prioritization & negotiations
-

Market-driven Software Development



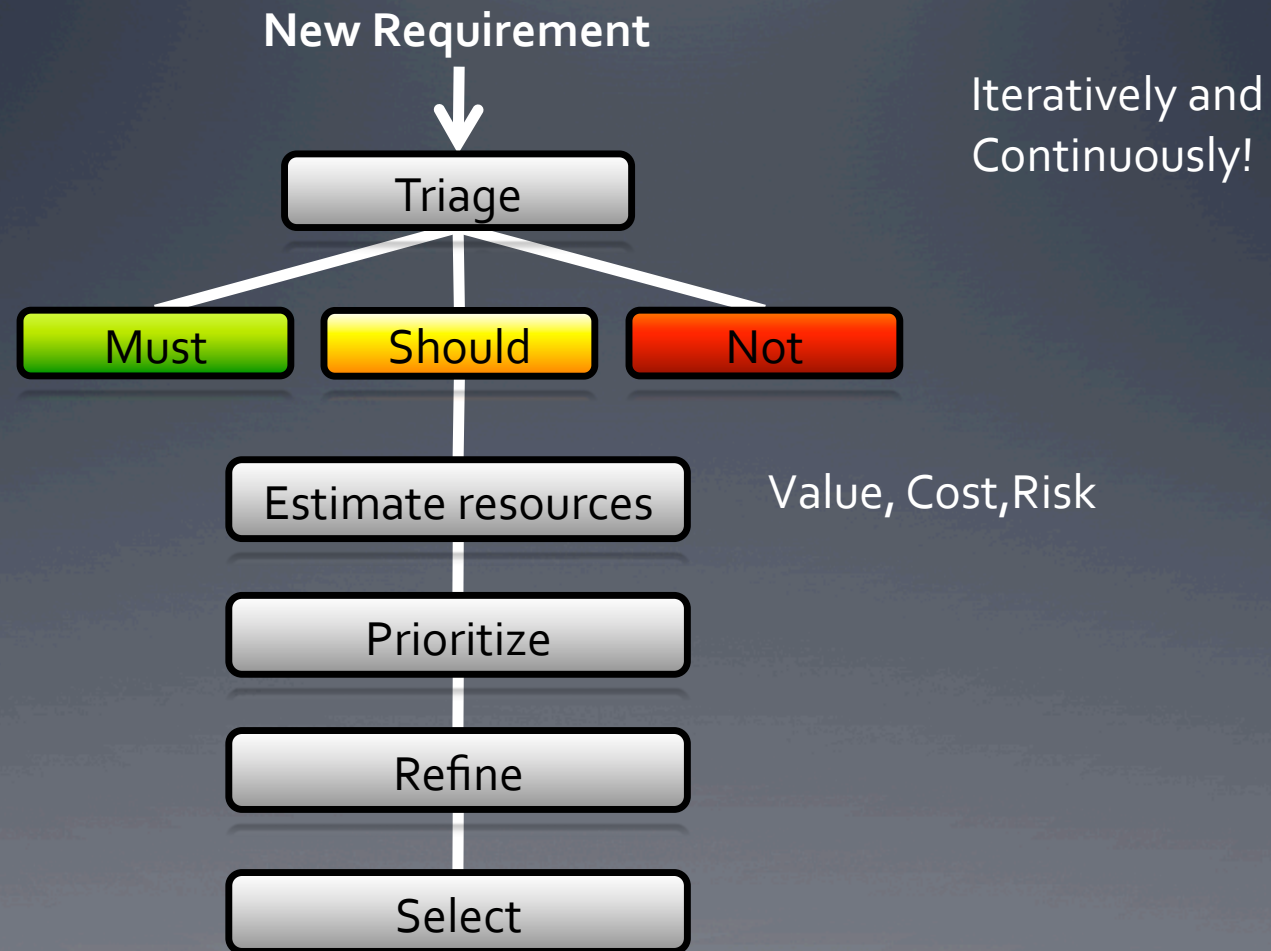
Market-Driven Software Development

- Many potential customers (companies and/or end Users)
 - No “negotiation”, rather elicitation, evaluation, prediction, innovation
 - Domain expertise primarily internally
 - Success = Sales volume, ROI, Market share, growth
-

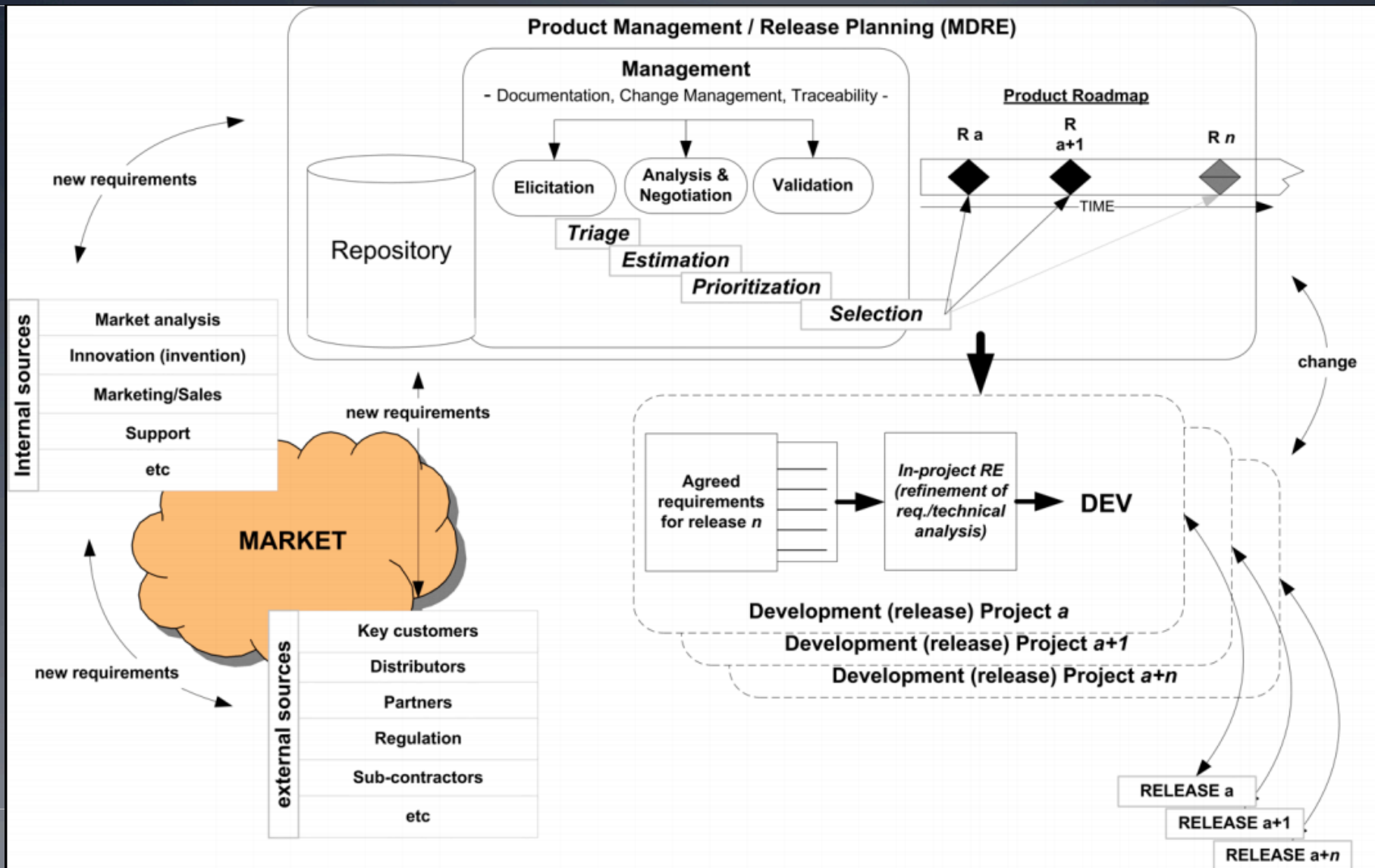
MD Software Development - RE steps

1. Decide **what business** you are in
 2. Select a **target market**
 3. **Market research** to determine size, competitors, customers, pains/needs, market message
 4. Draft high-level features in **Market Req Doc (MRD)** = desired price, intro date, prioritization
 5. **Test MRD** on potential customers
 6. Write a **detailed SRS**
 7. **Change** => internal triage/re-prioritize
 8. **Budget/Schedule problems** => internal triage/reprioritize
-

Requirement Triage (in MDRE)



MDRE (Gorschek2006, Khurum2009)



Industry Environment

Influences

Competition

Standards

Regulation

Trends

Stakeholders

Competitors

Distributors

Suppliers

Partners

End users

Sources

Market analysis

Marketing/sales

Segment analysis

Support

Focus groups

Management

Surveys

Experts

Developers

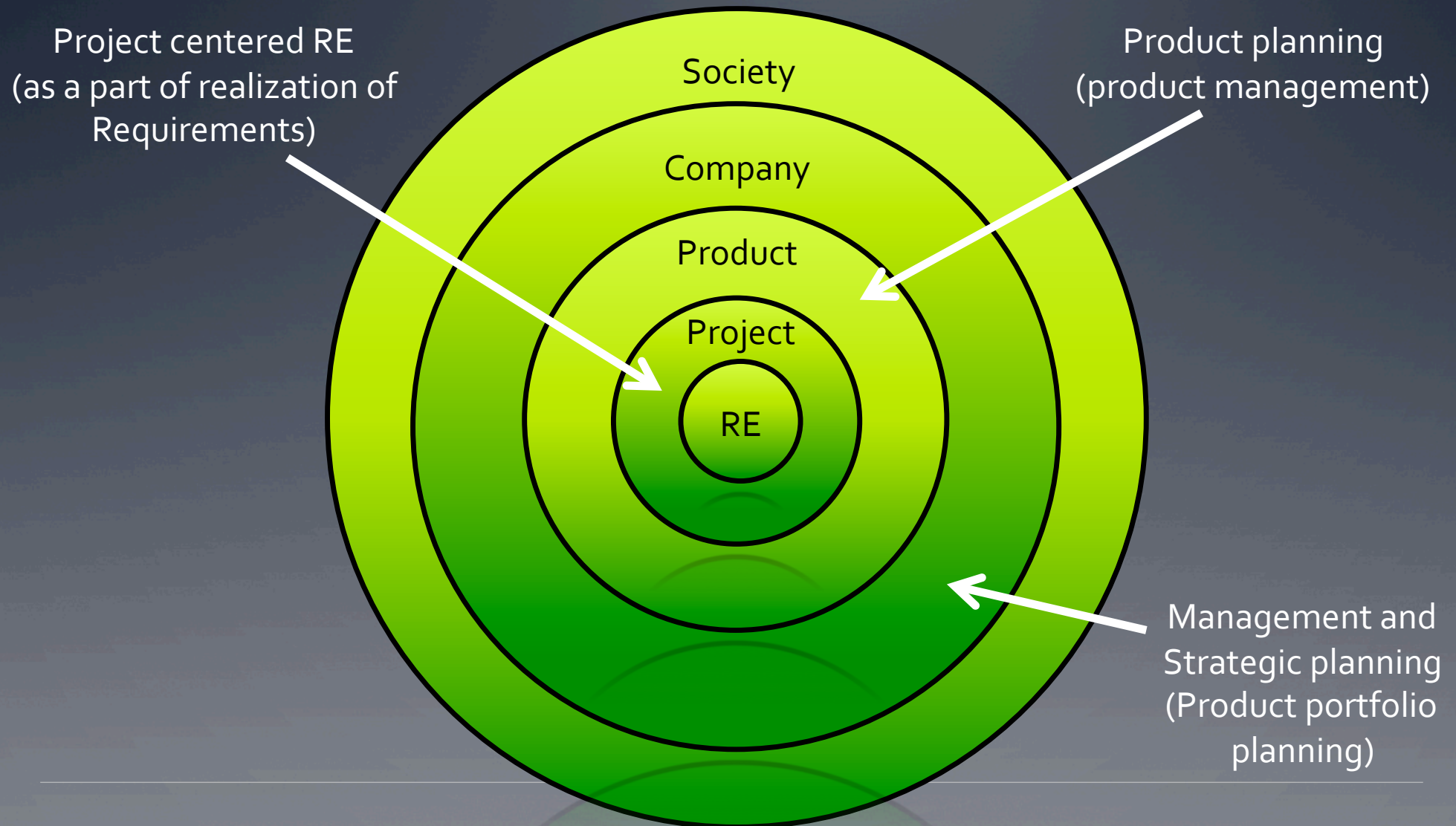
Competitor analysis

History

Industry Environment 2

- Large amount of information, data, requests, wishes, goals and requirements coming in all the time...
 - Limited only by when we choose to do cut-off
 - Multiple levels of abstraction and refinement
 - Traceability and access to requirement sources vary largely – e.g. getting hold of more information regarding req.
 - Multiple projects for each product
 - Multiple products for each company
-

Multiple perspectives of RE



Project perspective

- **Delivered to a project:**
 - A package of requirements
 - They are specified, initially analyzed and prioritized!
 - Project planning: estimations, initial analysis and risk analysis
-

Project perspective

- **Manage requirements** (V&V, refinement, update, risk analysis)
 - **Assure testability**
 - **Assure that the end-result** (e.g. software) of the project reflects the requirements allocated to the project
 - **Assure requirements**: Inspections, reviews
 - **Dependencies**
 - **Assure end-result** is in accordance with requirements: System test, acceptance tests, inspections, reviews
-

Product perspective

- Delivered to a product planning activity:
 - Company strategies
 - All product relevant requirements/ ideas/ data/ goals/ wishes
-

Product perspective (2)

- Explicit formulation of **product strategies**
 - Create **framework for requirements selection**
 - **Initial sorting of requirements** – e.g. with regards to product strategies
 - **Prioritization** (multiple levels – multiple perspectives)
 - **Dependencies** (initial)
 - **Package requirements** with regards to coupling and cohesion
 - Make **initial estimations** on req. and packages for project planning activities
 - Send packages to development projects for further **refinement** and realization
 - **Measure** how well the requirements selection process works
-

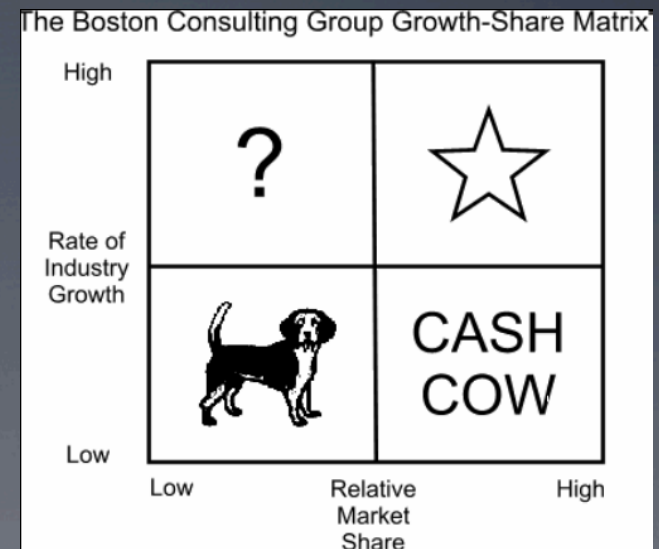
Product perspective (3)

- **Product planning success measurements** (wrt requirements):
 - GAP Analysis
 - Customer Value Analysis
 - Internal Value Analysis
 - Does the product generate revenue in accordance with estimations?
-

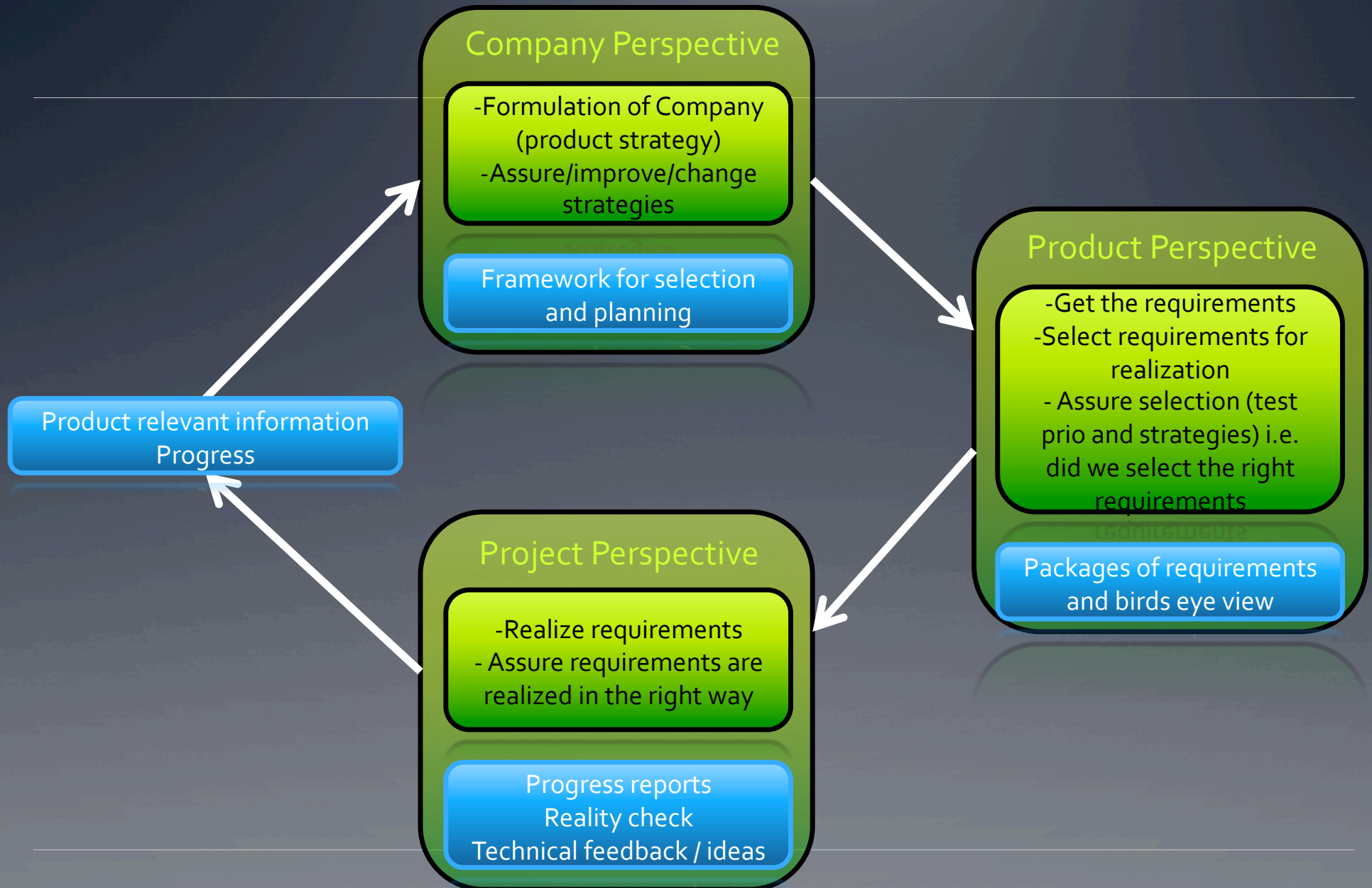
Company perspective

- Delivered to a company portfolio planning activity:
 - **Management relevant decision support data** (from development and product management, marketing, support etc)
 - **Input from efforts on Product Management** (e.g. GAP analysis, CVA, IVA etc)

- Output is:
 - Explicit formulation of **Company Strategies**
 - **Product strategies**
 - **Strategic development plans** with regards to several products (new, old... etc)
 - Tools used: Bubble diagrams



Overview



Requirements Overload

- **Large amounts of requirements:** Threat and opportunity
 - MDRE process needs to be able to handle large amounts of data continuously.
 - Overload can decrease quality if not handled correctly
-

Abstraction level

- **Goal like requirements** e.g. marketing channels, **to detailed technical solution** proposals from technically adapt customers.
 - **Requirements come in the raw form**. MDRE process needs to take this into consideration as it influences all aspects of later processing, whether it be early **triage, analysis and refinement, estimation, prioritization and ultimately selection**.
 - Process should be flexible enough to handle multiple types of requirements.
-

Requirements dependencies

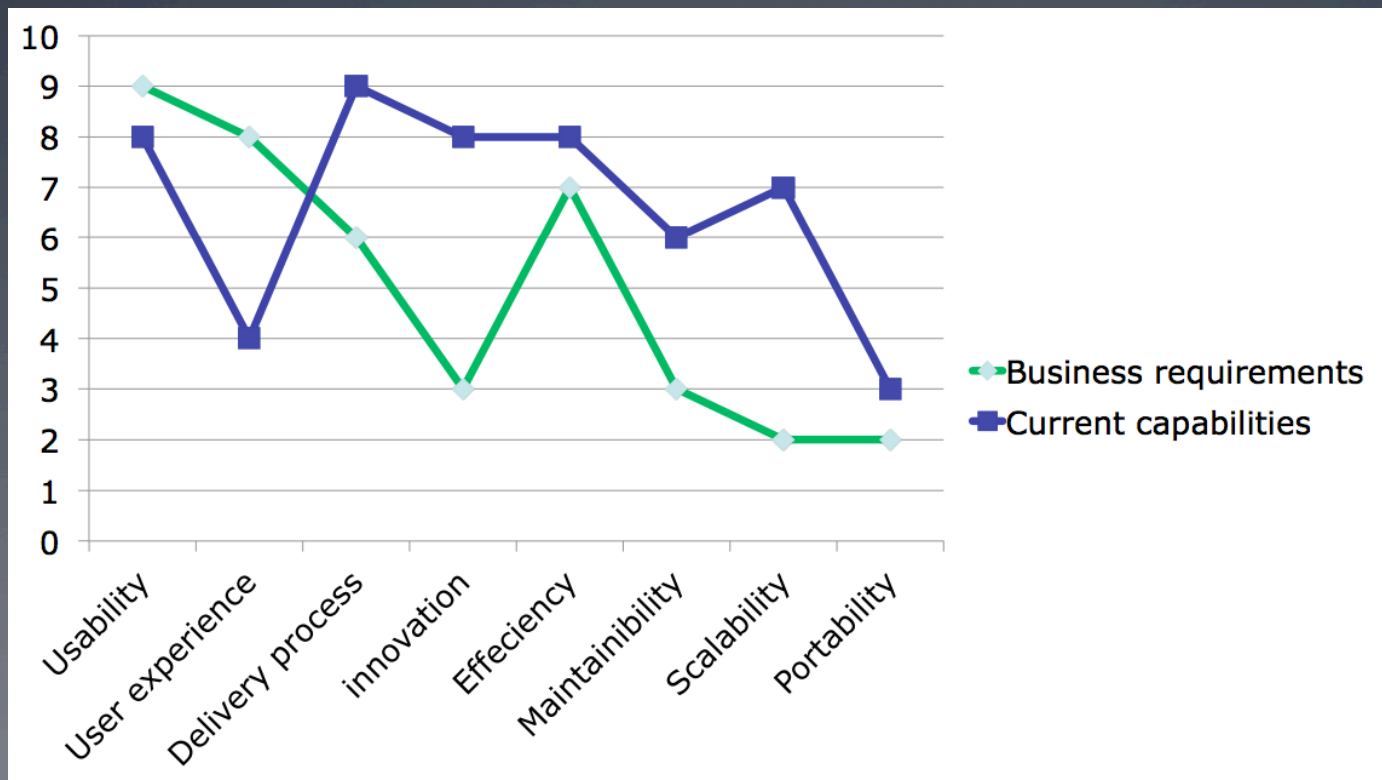
- Influence primarily requirements selection and release planning.
 - MDRE process needs to enable professionals to take dependencies into account when handling large amounts of requirements.
 - An important type of dependencies involves value-based dependencies, directly related to customer value and development cost.
-

Selection / Release planning

- Factors
 - Fixed Releases (time-to-market)
 - Estimation: Crucial
 - Prioritization
 - **Good-enough requirements** for estimation and prioritization (the consequence of a specific requirement needs to be known).
 - **This applies for prioritization** in order to better be able to compare. Similar abstraction level in addition to giving a good enough view of what they imply
-

Gap between market and product

- Positive gaps
- Negative gaps



Market pull / Technology push

- Types of requirements
 - Creating innovations (**technology-push**)
 - Requests/whishes/needs in the market environment (**market pull**).
 - Need to be balanced: use of product strategies (roadmaps) in requirement selection
-

Requirement Analysis

- Following early triage
 - Implementation costs and resources are estimated. Mostly ad-hoc estimation.
 - Time to market fixed.
 - Quality often sacrificed.
 - Value based dependencies, directly related to customer value and development cost important.
-

Requirement Prioritization

- Objective: prioritization for requirements selection for release planning.
 - • Success:
 - Outperforming competitors in the market
 - Delivering a high perceived benefit to customers. From this perspective customer satisfaction is central
 - Optimally customers (and potential customers) should perform prioritizations.
-

Requirement Prioritization

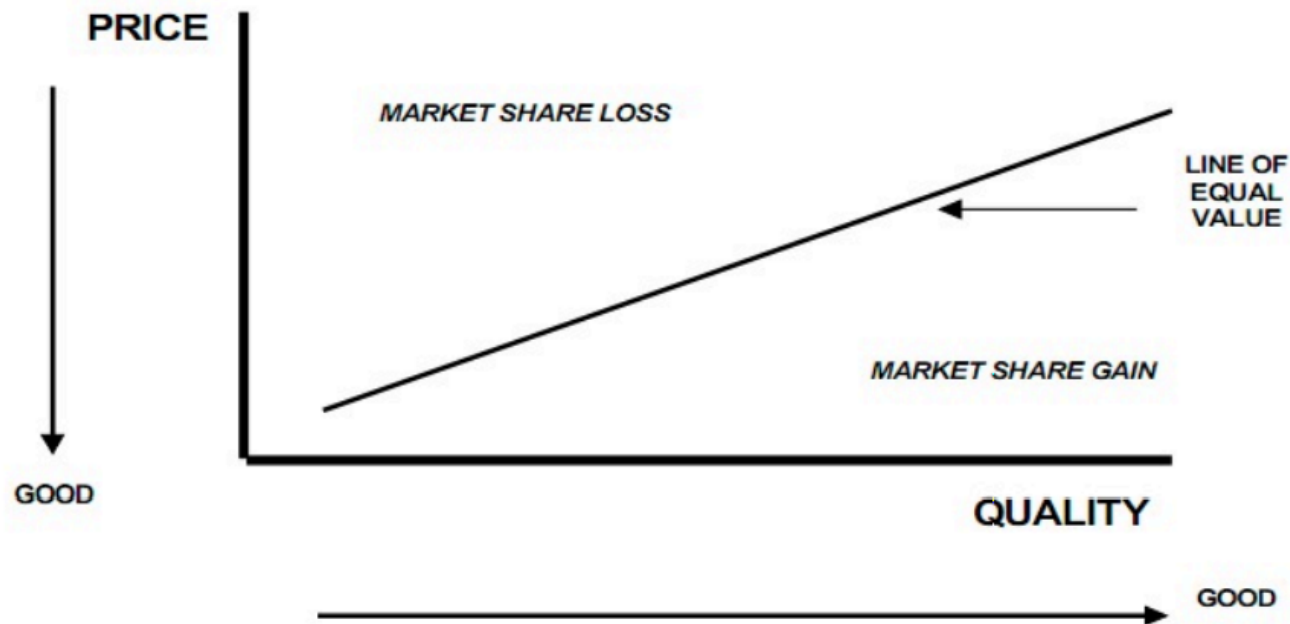
- Internal considerations regarding
 - **Technical aspects** (e.g. architecture and maintainability)
 - **Business aspects** (e.g. strategic decisions regarding focusing on new market-segments)
 - **Implementation aspects** (e.g. dependencies)
 - Several methods for attaining requirement priority exist, including the 100-point method, and the planning-game. [Scalability may be a problem]
-

Req. Selection (Release Planning)

- Use of roadmap to specify:
 - **Themes of a certain product release** (e.g. a theme could be offering a certain type of functionality, concentrating on improving quality, security and so on)
 - **Restrictions** (e.g. what are the restrictions in terms of risk, time, resources available, internal technical considerations and so on)
 - **Goals** (what are the overall product goals, and what are the goals for every release)
 - Milestones (for releases and goals)
 - **Balance technology-push and market-pull**
-

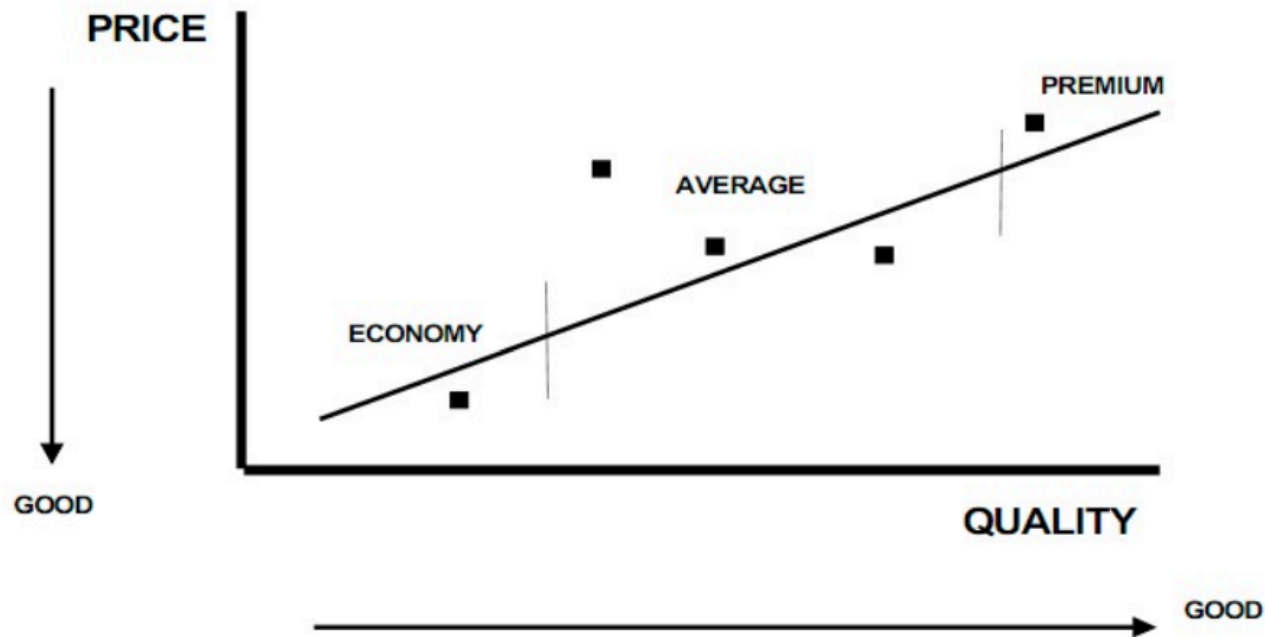
Customer Value Analysis

Value Map



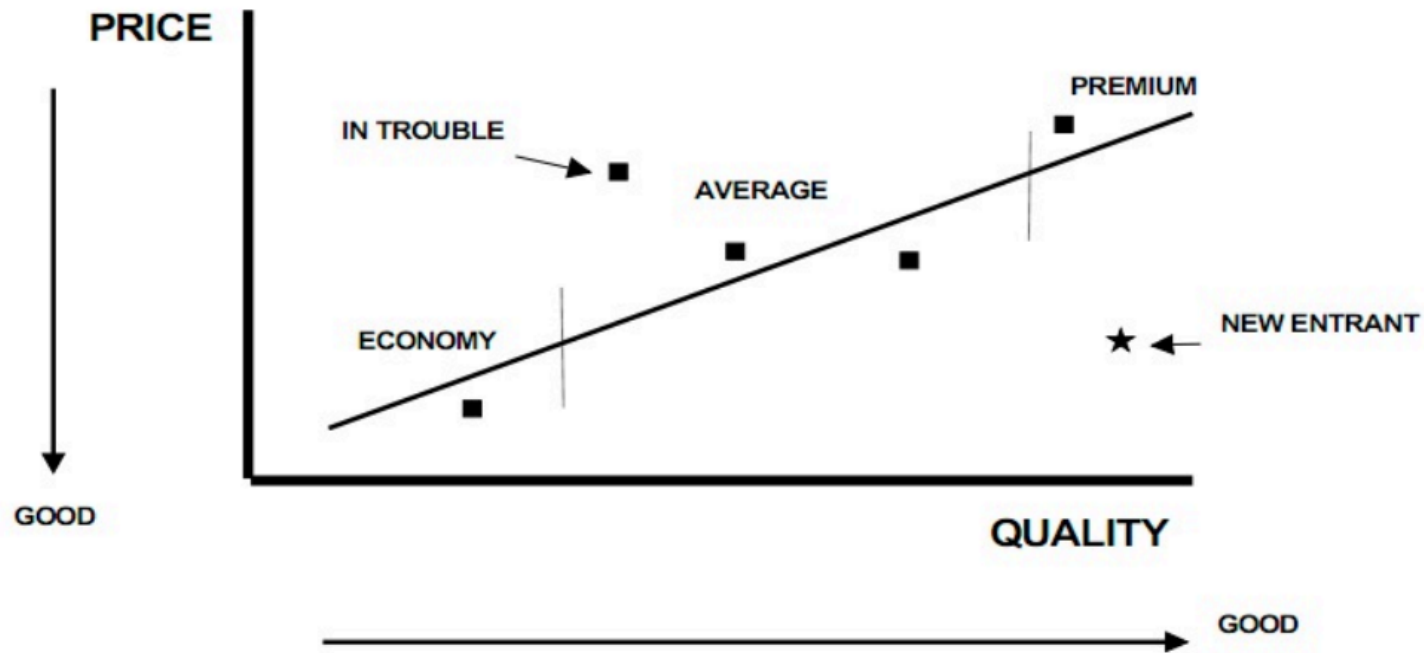
Customer Value Analysis

Typical Market “Before”



Customer Value Analysis

Typical Market “After”



Tools

- **Question:** Why should we use tools?
 - RE management is complex
 - Traceability
 - Change management
 - Communication
 - Etc.
 - Specialized versus Standard office application tools
 - **Specialized:** More support, harder to learn
 - **Office applications:** Little support, requires process but easy to learn
 - Different tools:
 - DOORs, CaliberRM, Optimal Trace, Rational Requisite Pro, etc
-

Properties of a RM tool

- Manage different information
 - Manage logical relationships between information
 - Allow for unique identification
 - Edit managed information
 - Provide different views on managed information
 - Organize information
 - Generate reports or summaries
 - Generate different output
-

Tool Evaluation

- There are many tools, which should we use?
 - Objective, multi-perspective/view analysis
 - **Project view** – Project support, planning, etc
 - **User view** – Support different roles, distributed work, etc
 - **Product view** – Different documentation support, etc
 - **Process view** – Extend tool with method guidance, etc
 - **Provider view** – Tool provider reputation, support, etc
 - **Technical view** – Performance, scalability, etc. of the tool
 - **Economic view** – Acquisition, introduction, maintenance costs
-

DOORS

- Comprehensive and widely used
 - **Views to identify**
 - Areas of change
 - Traceability to identify risks
 - Point-and-click views for different roles
 - Support to identify areas of change (Alignment between requirements and tests)
 - **Requirements change** management (individual reqs.)
 - **Lifecycle change** management (Impact on product)
 - **Change proposal system**
 - Group and manage change
-

Questions?
