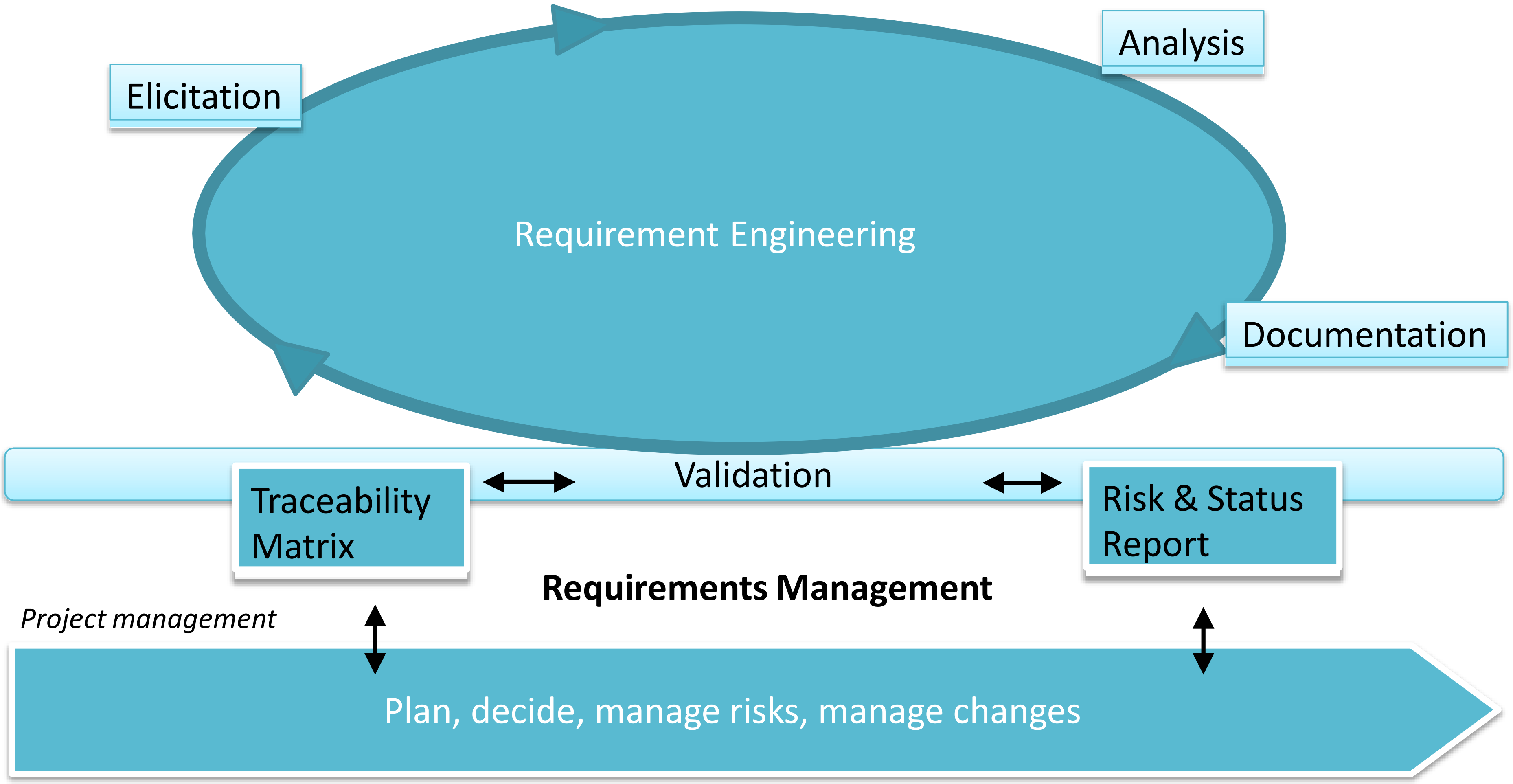
Risk Analysis and Management

Requirements Engineering

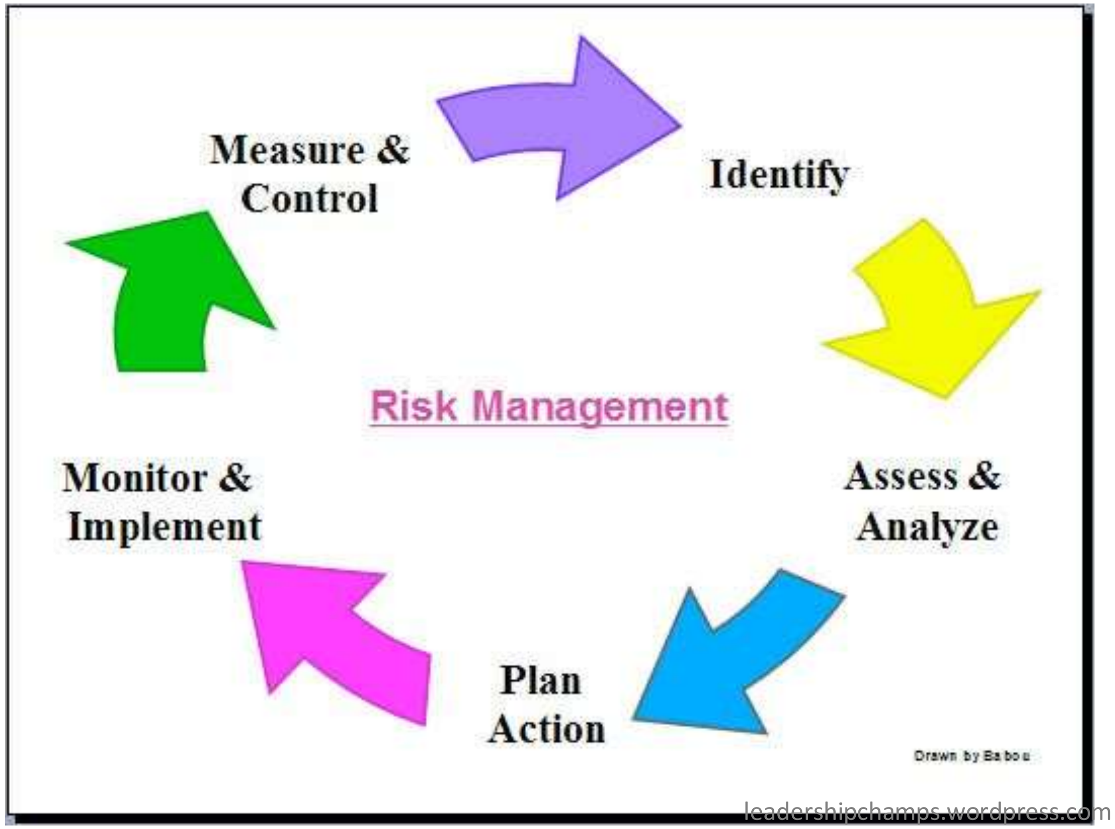
Recap: RE in the process



Risk managment
Fundmanetals of RM
->elements
   -->activities
->document
->planning
->Tools
   --> swot
   -->root cause
       -->fish bone
Req related RM
 ->eliciation
 ->analysis
 ->specification
 ->validation
 ->managment
RM is your phrand
Risk reduction through prototyping
 ->intro
 -> types
 ->working
 ->success factor

Risk Management

Risks that are related to requirements have to be identified, assessed and analyzed, so that action can be taken early on.

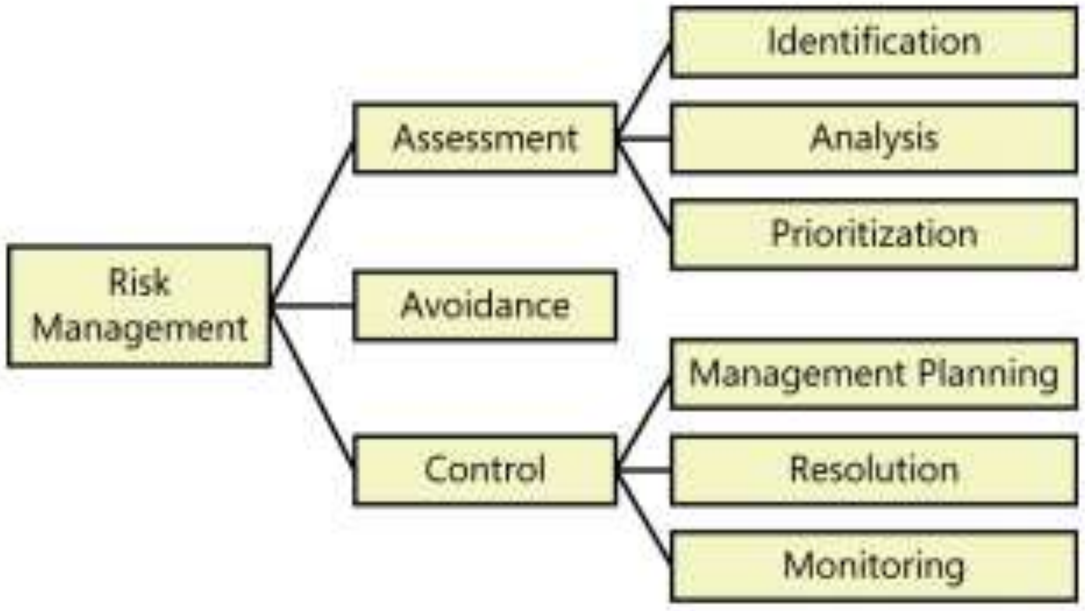


leadershipchamps.wordpress.com

5 steps of risk mngmnt

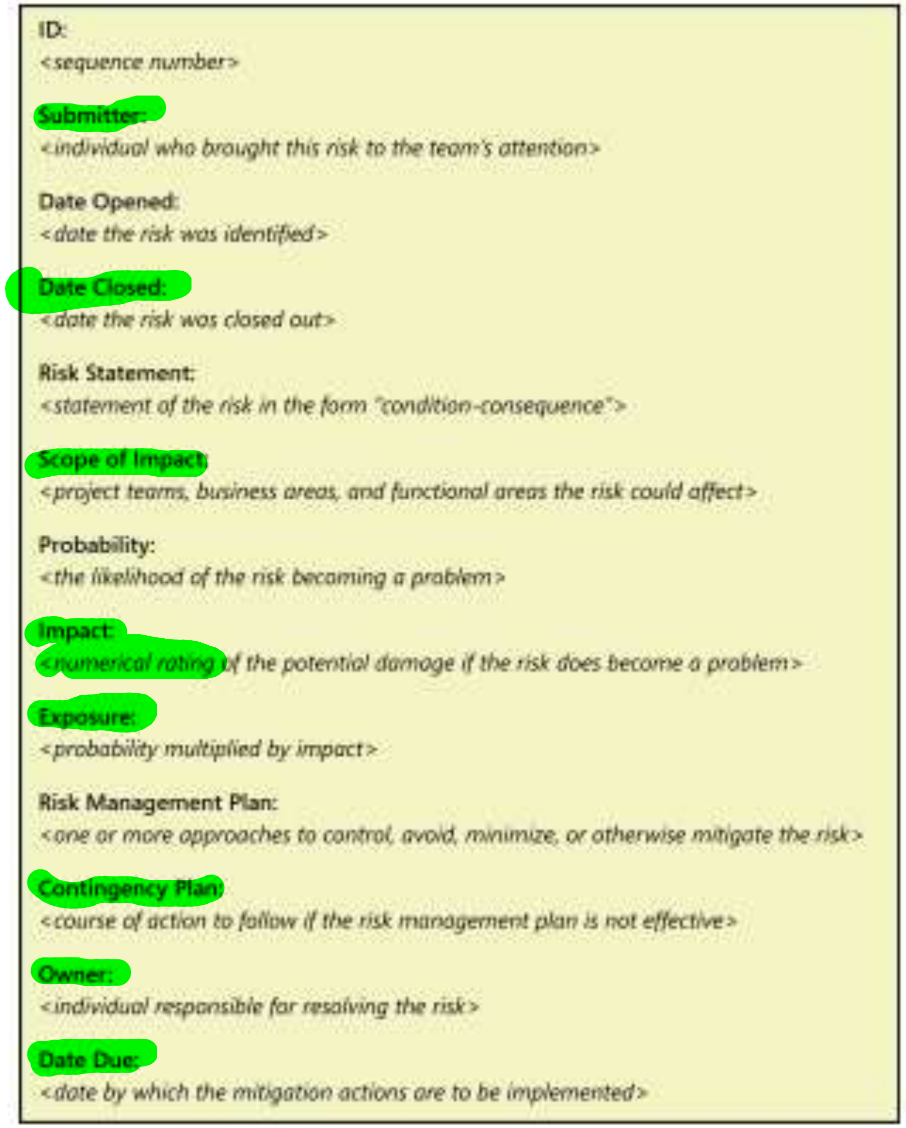
Fundamentals of software risk management Elements of risk management

- Risk management involves the application of tools and procedures to contain project risk within acceptable limits.
- It provides a standard approach to identify and document risk factors, evaluate their potential severity, and propose strategies for mitigating them
- Risk management includes the activities shown below



Documenting project risks

- It’s not enough to simply recognize the risks that face your project.
- You need to manage them in a way that lets you communicate risk issues and status to stakeholders throughout the project’s duration.
- Keep the risk list separate from project plans so that it’s easy to update throughout the project’s duration.



ID:
<sequence number>

Submitter:
<individual who brought this risk to the team’s attention>

Date Opened:
<date the risk was identified>

Date Closed:
<date the risk was closed out>

Risk Statement:
<statement of the risk in the form "condition-consequence">

Scope of Impact:
<project teams, business areas, and functional areas the risk could affect>

Probability:
<the likelihood of the risk becoming a problem>

Impact:
<numerical rating of the potential damage if the risk does become a problem>

Exposure:
<probability multiplied by impact>

Risk Management Plan:
<one or more approaches to control, avoid, minimize, or otherwise mitigate the risk>

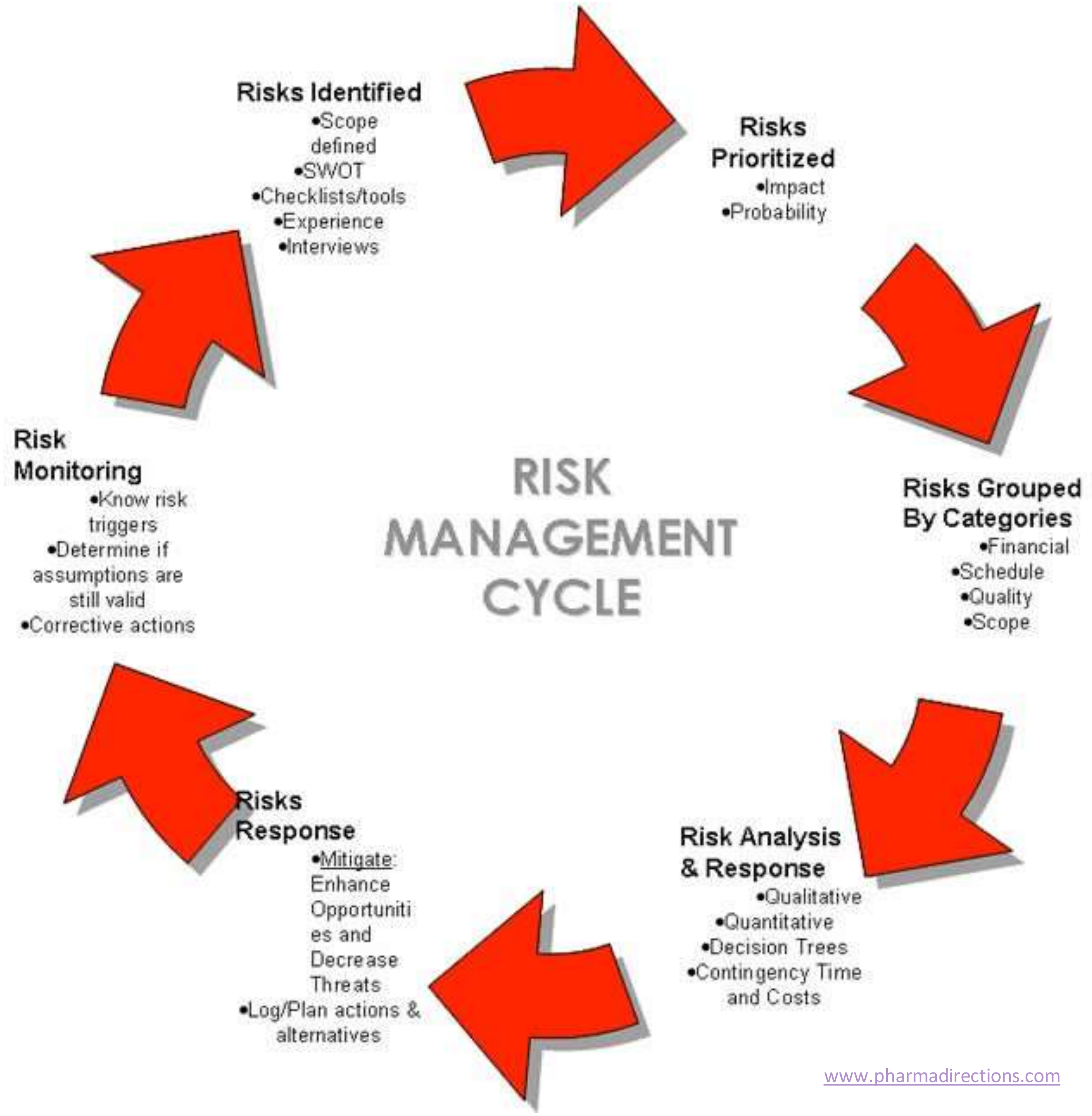
Contingency Plan:
<course of action to follow if the risk management plan is not effective>

Owner:
<individual responsible for resolving the risk>

Date Due:
<date by which the mitigation actions are to be implemented>

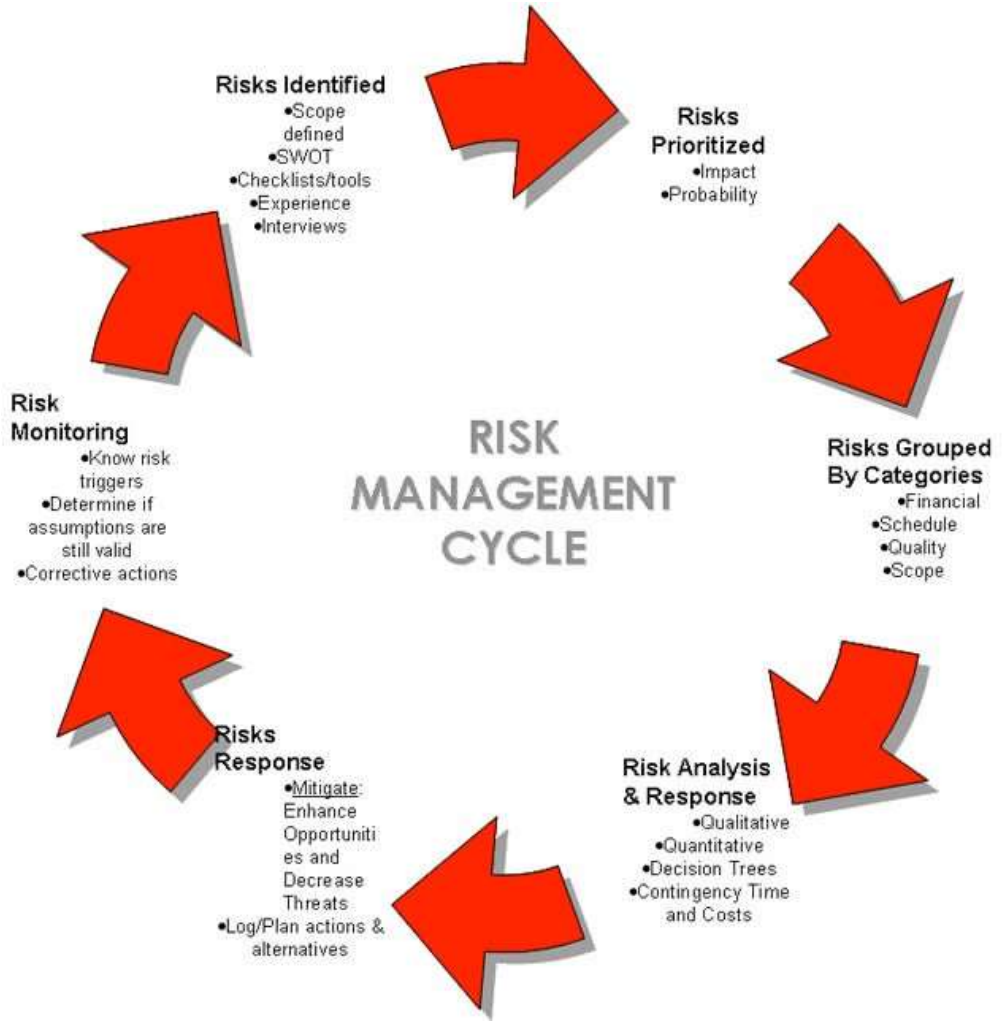
Planning for risk management

- For a small project, you can include your plans for controlling risks in the software project management plan.
- A large project should write a separate risk management plan that spells out the approaches it intends to take to identify, evaluate, document, and track risks.
- This plan should include the roles and responsibilities for the risk management activities.
- Keep periodic risk monitoring of 0 or so risks that have the highest risk exposure visible, and track the effectiveness of your mitigation approaches regularly.
- When a mitigation action is completed, reevaluate the probability and impact for that risk item and then update the risk list and any other pending mitigation plans accordingly.
- A risk is not necessarily under control simply because the mitigation actions have been completed. You need to judge whether your mitigation approaches have reduced the exposure to an acceptable level or whether the opportunity for a specific risk to become a problem has passed.

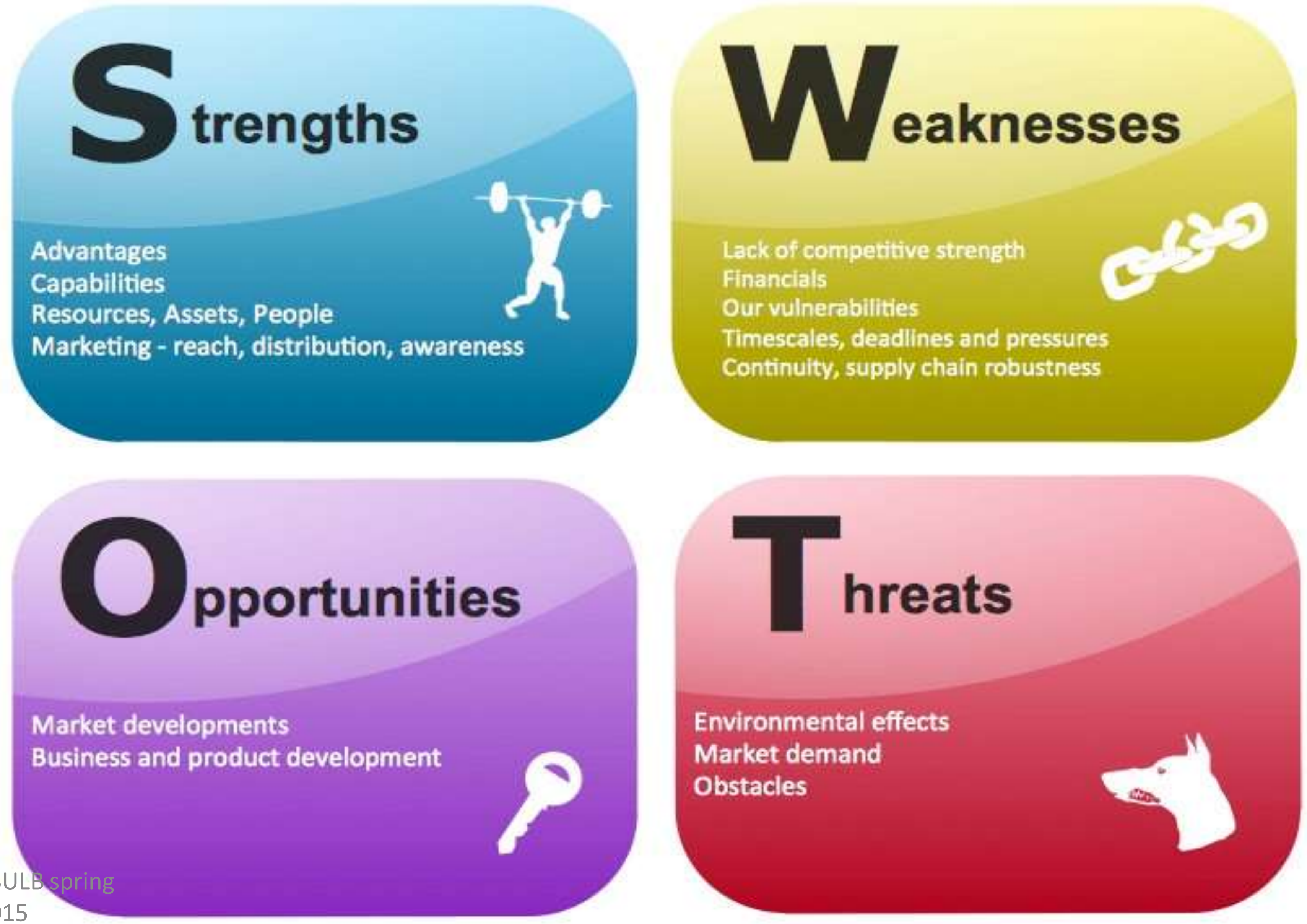

Risks Identified
•Scope defined
•SWOT
•Checklists/tools
•Experience
•Interviews
Risks Prioritized
•Impact
•Probability
Risks Grouped By Categories
•Financial
•Schedule
•Quality
•Scope
Risk Analysis & Response
•Qualitative
•Quantitative
•Decision Trees
•Contingency Time and Costs
Risks Response
•Mitigate: Enhance Opportunities and Decrease Threats
•Log/Plan actions & alternatives
Risk Monitoring
•Know risk triggers
•Determine if assumptions are still valid
•Corrective actions
RISK MANAGEMENT CYCLE

Risk Management Tools

- SWOT analysis (Strengths, weaknesses, opportunities, threats)
- Root-cause analysis



SWOT analysis



Strengths

Advantages
Capabilities
Resources, Assets, People
Marketing - reach, distribution, awareness

Weaknesses

Lack of competitive strength
Financials
Our vulnerabilities
Timescales, deadlines and pressures
Continuity, supply chain robustness

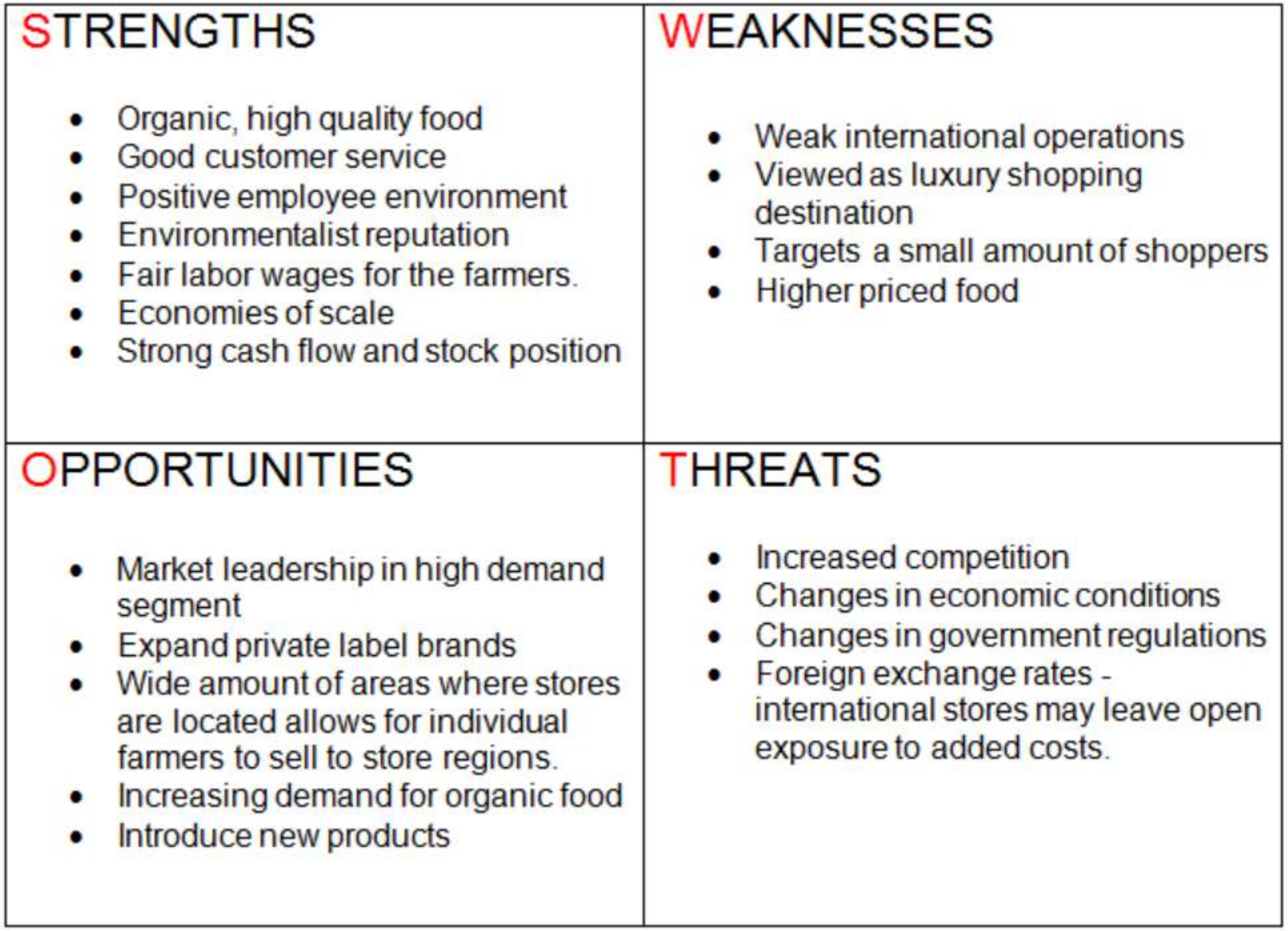
Opportunities

Market developments
Business and product development

Threats

Environmental effects
Market demand
Obstacles

SWOT – Whole Food Market



STRENGTHS

- Organic, high quality food
- Good customer service
- Positive employee environment
- Environmentalist reputation
- Fair labor wages for the farmers.
- Economies of scale
- Strong cash flow and stock position

WEAKNESSES

- Weak international operations
- Viewed as luxury shopping destination
- Targets a small amount of shoppers
- Higher priced food

OPPORTUNITIES

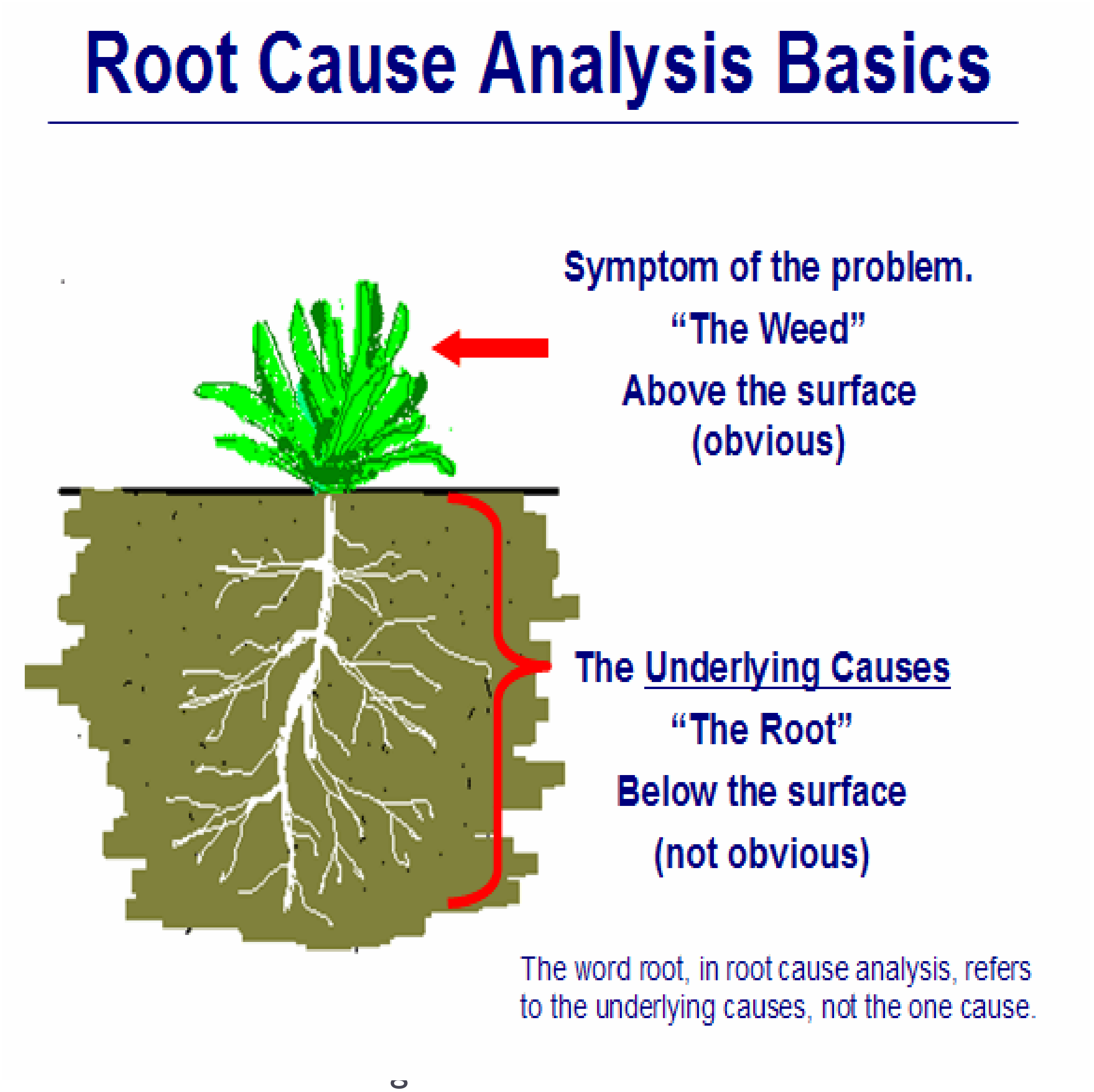
- Market leadership in high demand segment
- Expand private label brands
- Wide amount of areas where stores are located allows for individual farmers to sell to store regions.
- Increasing demand for organic food
- Introduce new products

THREATS

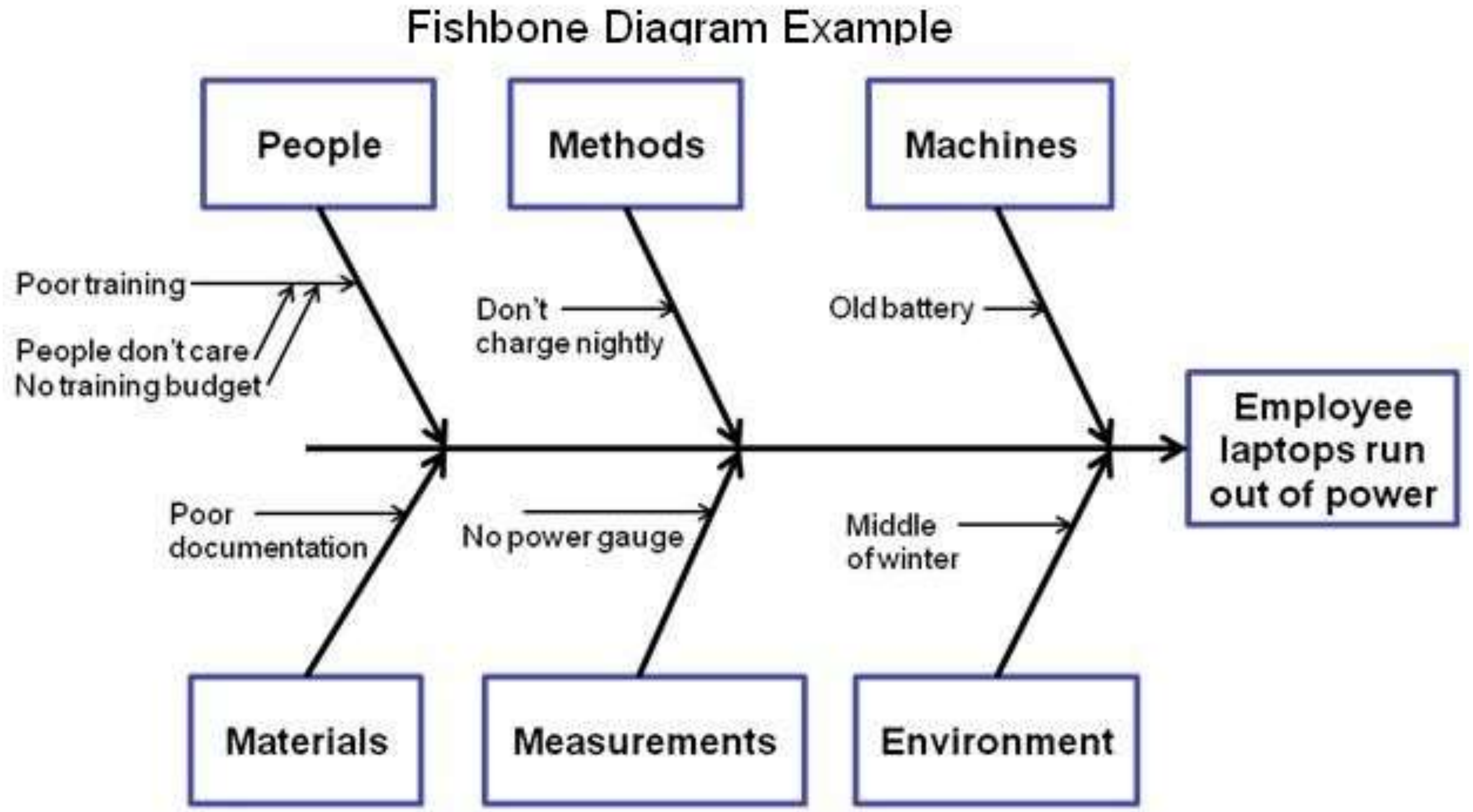
- Increased competition
- Changes in economic conditions
- Changes in government regulations
- Foreign exchange rates - international stores may leave open exposure to added costs.

Root-cause analysis

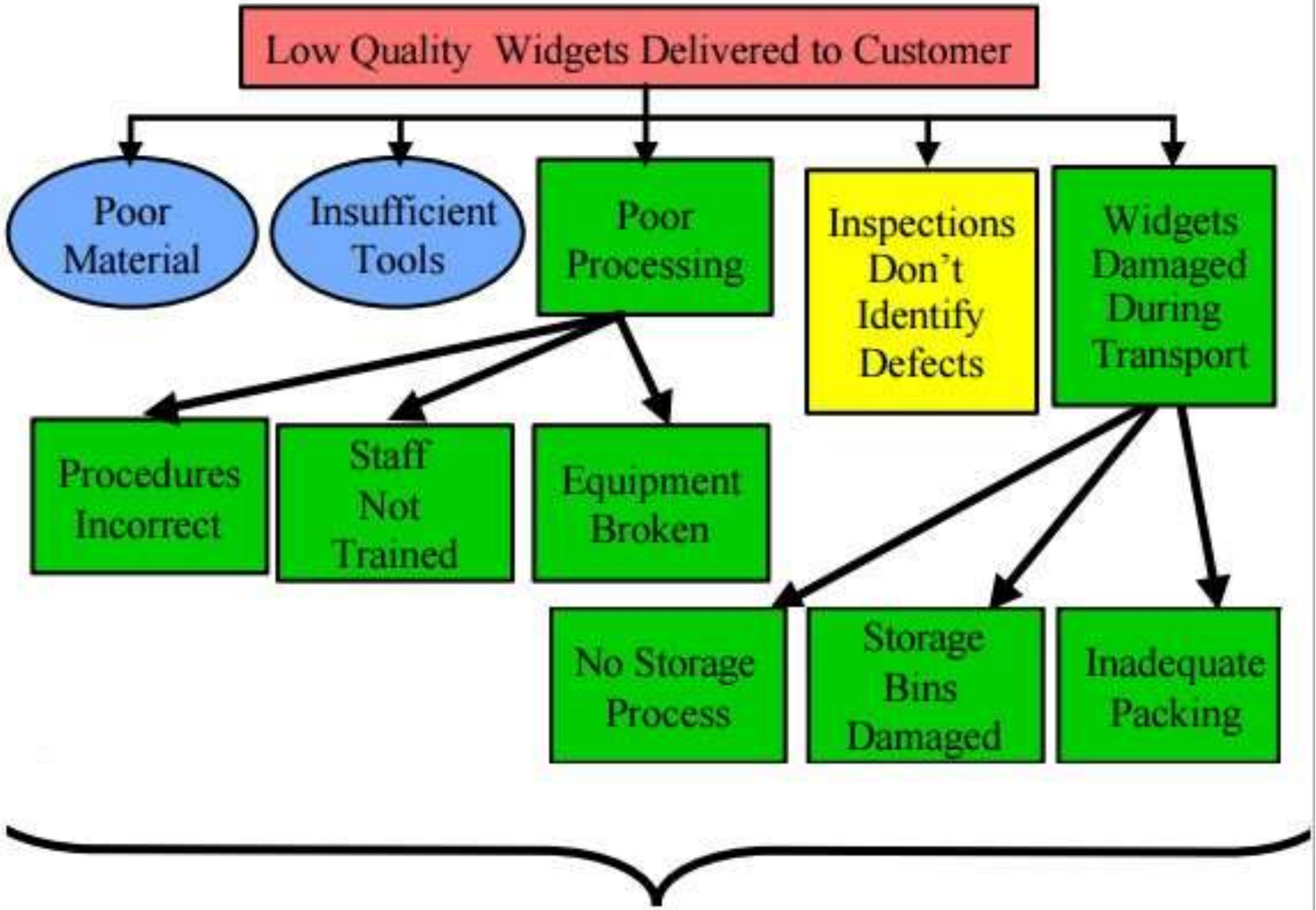
- (Strengths, weaknesses, opportunities, threats)
- Root-cause analysis



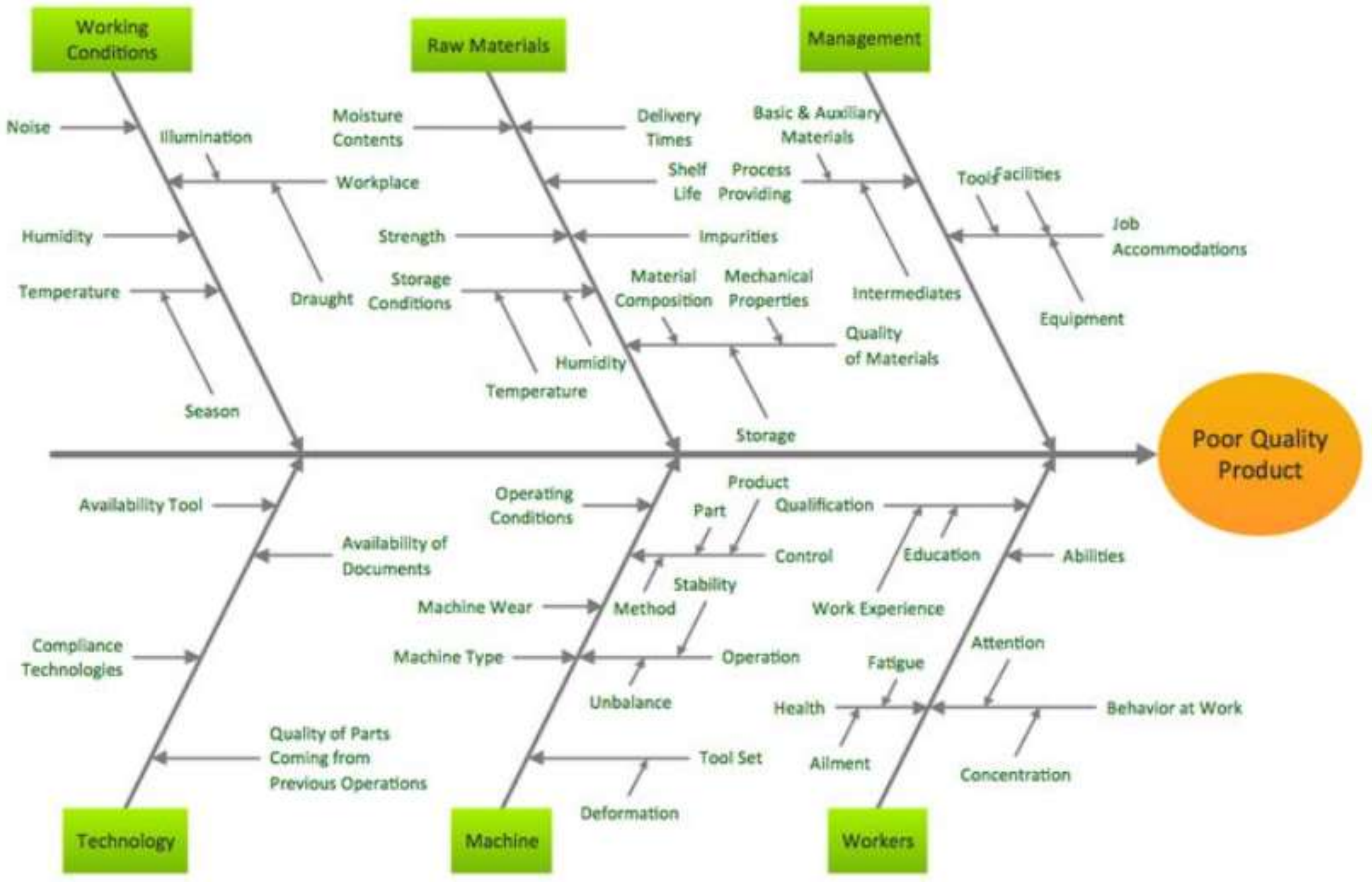
Root-cause analysis



Root Cause Analysis - Example



Root-cause analysis



Requirements-related risks

Requirements elicitation Numerous factors can conspire to hamper your requirements elicitation efforts. Following are several areas of potential elicitation risk.

- Product vision and project scope
- Time spent on requirements development
- Customer engagement
- Completeness and correctness of requirements specifications
- Requirements for innovative products
- Defining nonfunctional requirements
- Customer agreement on requirement
- Unstated requirements
- Existing product used as the requirements reference
- Solutions presented as needs
- Distrust between the business and the development team

Requirements-related risks

Requirements analysis

It isn’t prudent to just record whatever the customer tells you and dive into development. Requirements analysis poses its own threat areas, as described below.

- Requirements prioritization
- Technically difficult features
- Unfamiliar technologies, methods, languages, tools, or hardware

Requirements-related risks

Requirements specification

Requirements are all about communication. Just because requirements are communicated on paper or in writing doesn't mean they are actually understood.

- Requirements understanding
- Time pressure to proceed despite open issues
- Ambiguous terminology
- Design included in requirements

Requirements-related risks

Requirements validation

Even if you’ve done a good job on requirements elicitation, it’s important to confirm the quality and validity of the solution that the requirements specify. Validation offers the following pitfalls.

- Invalidated requirements
- Inspection proficiency

Requirements-related risks

Requirements management

Much of the requirements-related risk on a software project comes from how changes are handled. Those and other requirements management risks are mentioned below.

- Changing requirement
- Requirements change process
- Unimplemented requirements
- Expanding project scope

Risk management is your friend

- A project manager can use risk management to raise the awareness of conditions that could cause the project to suffer.
- Consider the manager of a new project who's concerned about getting appropriate users involved in requirements elicitation.
- The astute manager will realize that this condition poses a risk and will document it in the risk list, estimating the probability and impact based on previous experience.
- If time passes and users still are not involved, the risk exposure for this item will increase, perhaps to the point where it compromises the project's success.
- I've been able to convince managers to postpone a project that could not engage sufficient user representatives by arguing that we shouldn't waste the company's money on a doomed project.
- Periodic risk tracking keeps the project manager apprised of the threat from identified risks.
- Escalate risks that aren't adequately controlled to senior managers, who can either initiate corrective actions or make a conscious business decision to proceed despite the risks.
- Risk management helps you keep your eyes open and make informed decisions, even if you can't control or avoid every adversity your project might encounter.

Risk reduction through prototyping

A software prototype is a partial, possible, or preliminary implementation of a proposed new product. Prototypes has three main purpose:

- Clarify, complete, and validate requirements
- Explore design alternatives
- Create a subset that will grow into the ultimate product

The primary reason for creating a prototype is to resolve uncertainties early in the development process. Focus on high-risk areas or known uncertainties to decide which parts of the system to prototype. A prototype is useful for revealing and resolving ambiguity and incompleteness in the requirements.

classes of prototype

- Scope

  A mock-up prototype focuses on the user experience; a proof-of-concept prototype explores the technical soundness of a proposed approach.

- Future use

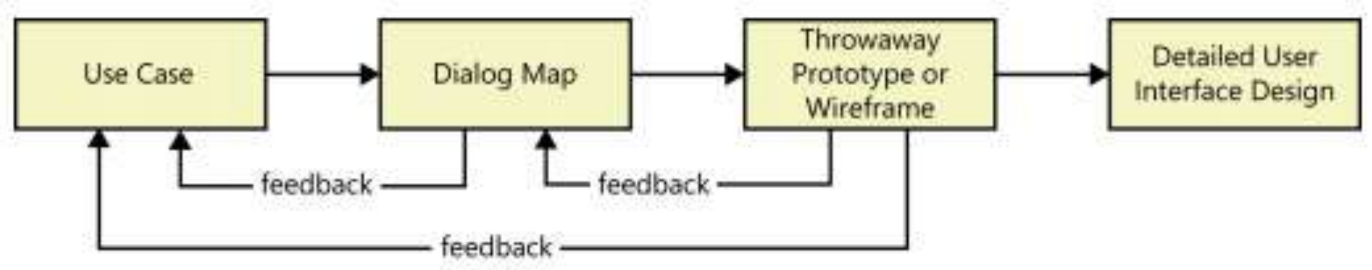
  A throwaway prototype is discarded after it has been used to generate feedback, whereas an evolutionary prototype grows into the final product through a series of iterations.

- Form

  A paper prototype is a simple sketch drawn on paper, a whiteboard, or in a drawing tool. An electronic prototype consists of working software for just part of the solution.

Working with prototypes

Activity sequence from use cases to user interface design using a throwaway prototype.


Use Case
Dialog Map
Throwaway Prototype or Wireframe
Detailed User Interface Design
feedback
feedback
feedback

Prototyping success factors

Software prototyping provides a powerful set of techniques that can minimize development schedules, ensure customer satisfaction, and produce high-quality products. To make prototyping an effective part of your requirements process, follow these guidelines.

- Include prototyping tasks in your project plan. Schedule time and resources to develop, evaluate, and modify the prototypes.
- State the purpose of each prototype before you build it, and explain what will happen with the outcome:
- Plan to develop multiple prototypes. You'll rarely get them right on the first try, which is the whole point of prototyping!

Prototyping success factors

- Create throwaway prototypes as quickly and cheaply as possible with the minimum amount of effort
- Don’t include input data validations or extensive code documentation in a throwaway prototype.
- Don’t prototype requirements that you already understand, except to explore design alternatives.
- Use conceivable data in prototype screen displays and reports
- Don’t expect a prototype to replace written requirements. Screen images don’t give the details of data field definitions and validation criteria, relationships between fields, exception handling, business rules, and other essential bits of information.