Risk Management

Peter Dolog
dolog [at] cs [dot] aau [dot] dk
5.2.47
Information Systems
March 26, 2008

What is Risk Management?

Making informed decisions by consciously assessing what can go wrong and the severity of its impact.

Risk*:

- Is an undesirable event
- Involves uncertainty - the probability the event will occur
- Involves loss - the impact should the event occur

* Note: Risk is the possibility of suffering loss, injury, disadvantage, or destruction [Webster's Dictionary 81, p. 1961]

Software Risk Program, Ron Higuera, Software Engineering Institute

Risk management

Perhaps the principal task of a manager is to minimize risk

The 'risk' inherent in an activity is a measure of the uncertainty of the outcome of that activity

High-risk activities cause schedule and cost overruns

Risk is related to the amount and quality of available information. The less information, the higher the risk

Pressman, 2005

Project Risks

What can go wrong?
What is the likelihood?
What will the damage be?
What can we do about it?

Pressman, 2005

Seven Principles

Maintain a global perspective—view software risks within the context of system and the business problem

Take a forward-looking view—think about the risks that may arise in the future; establish contingency plans

Emphasize a continuous process—the team must be vigilant throughout the software process, modifying identified risks as more information is known and adding new ones as better insight is achieved.

Integrate—a consideration of risk must be integrated into the software process

Encourage open communication—if someone states a potential risk, don't discount it.

Develop a shared product vision—if all stakeholders share the same vision of the software, it is likely that better risk identification and assessment will occur.

Encourage teamwork—the talents, skills and knowledge of all stakeholder should be pooled

Pressman, 2005

Mitigation, Monitoring, Management

Mitigation—how can we avoid the risk?

- Mitigate - to lessen in force or intensity; to make less severe; to make milder or more gentle

Monitoring—what factors can we track that will enable us to determine if the risk is becoming more or less likely?

- Monitor - to listen to; to view or listen; to observe, record, or detect an operation or condition; to observe critically, oversee, supervise

Management—what contingency plans do we have if the risk becomes a reality?

[Webster's Encyclopedic Unabridged Dictionary of the English Language, 1996] Pressman, 2005

Risk Identification

A systematic attempt to specify threats to the project plan.

- avoid when possible and control when necessary.

Two types of risks:

- Generic Risks
- Product-Specific Risks

One method for identifying risks is to create a risk item checklist which can be used for risk identification.

Pressman, 2005

Kinds of Software Risks

Project Risks threaten the project plan.

Business Risks threaten the viability of the software to be built.

Product or Technical Risks threaten the quality and timeliness of the software to be produced.

Known Risks are those that can be uncovered after careful evaluation

Predictable Risks are extrapolated from past project experience

Unpredictable Risks can and do occur.

Pressman, 2005

Risk Item Checklists

Product size.

Business impact.

Customer characteristics.

Process definition.

Development environment.

Technology to be built.

Staff size and experience.

Pressman, 2005

Top 10 software risks

Personnel Shortfalls

Unrealistic schedules and budgets

Developing the wrong functions and properties

Developing the wrong user interface

Gold - plating

Continuing stream of requirements changes

Shortfalls in externally furnished components

Shortfalls in externally performed tasks

Real - time performance shortfalls

Straining computer - science capabilities

Boehm, 1991

Product Size Risks

- Estimated size of the product in FP or LOC?

  Degree of confidence in size estimate?

- Estimated size of product in number of programs, files, transactions?

  Percentage deviation in size of product from average for previous products?

- Size of database created or used by the product?

  Number of users of the product?

- Number of projected changes to the requirements for the product? before delivery? after delivery?

- Amount of reused software?

Pressman, 2005

Business Impact Risks

- Effect of this product on company revenue?
- Visibility of this product to senior management?
- Reasonableness of delivery deadline?
- Number of customers who will use this product and the consistency of their needs relative to the product?
- Number of other products/systems with which this product must be interoperable?

Pressman, 2005

Business Impact Risks

- Sophistication of end users?
- Amount and quality of product documentation that must be produced and delivered to the customer?
- Governmental constraints on the construction of the product?
- Costs associated with late delivery?
- Costs associated with a defective product?

Pressman, 2005

Customer Related Risks

- Have you worked with the customer in the past?
- Does the customer have a solid idea of what is required? Has the customer spent the time to write it down?
- Will the customer agree to spend time in formal requirements gathering meetings to identify project scope?
- Is the customer willing to establish rapid communication links with the developer?
- Is the customer willing to participate in reviews?
- Is the customer technically sophisticated in the product area?
- Is the customer willing to let people do their job - that is, will the customer resist looking over your shoulder during technically detailed work?
- Does the customer understand the software process?

Pressman, 2005

Process & Technical Issues

Process Issues

- Does your senior management support a written policy statement that emphasizes the importance of a standard process for software development?
- Has your organization developed a written description of the software process to be used on this project?
- Are staff members 'signed up' to the software process as it is documented and willing to use it?
- Are CASE tools used for analysis, design and testing

Technical Issues

- Are facilitated application specification techniques used to aid in communication between the customer and developer?
- Are specific methods used for software analysis?
- Do you use a specific method for data and architectural design?
- Is more than 90 percent of your code written in a high-order language?
- Are specific conventions for code documentation defined and used?
- Do you use specific methods for test case design?

Pressman, 2005

Technology Risk

- Is the technology to be built new to your organization?
- Do the customer's requirements demand the creation of new algorithms or input or output technology?
- Does the software interface with new or unproven hardware?
- Does the software to be built interface with vendor supplied software products that are unproven?
- Does the software to be built interface with a database system whose function and performance have not been proven in this application area?
- Is a specialized user interface demanded by product requirements?

Pressman, 2005

Development Environment Risk

- Is a software project management tool available?
- Is a software process management tool available?
- Are tools for analysis and design available?
- Do analysis and design tools deliver methods that are appropriate for the product to be built?
- Are compilers or code generators available and appropriate for the product to be built?
- Are testing tools available and appropriate for the product to be built?
- Are software configuration management tools available?

Pressman, 2005

Staff Size and Experience

- Are the best people available?
- Do the people have the right combination of skills?
- Are enough people available?
- Are staff committed for entire duration of the project?
- Will some project staff be working only part time on this project?
- Do staff have the right expectations about the job at hand?
- Have staff received necessary training?
- Will turnover among staff be low enough to allow continuity?

Pressman, 2005

Observations

Most managers believe they are managing risk.

Risk is seldom managed systematically.

Managers tend to address only risks from technical areas with which they have expertise.

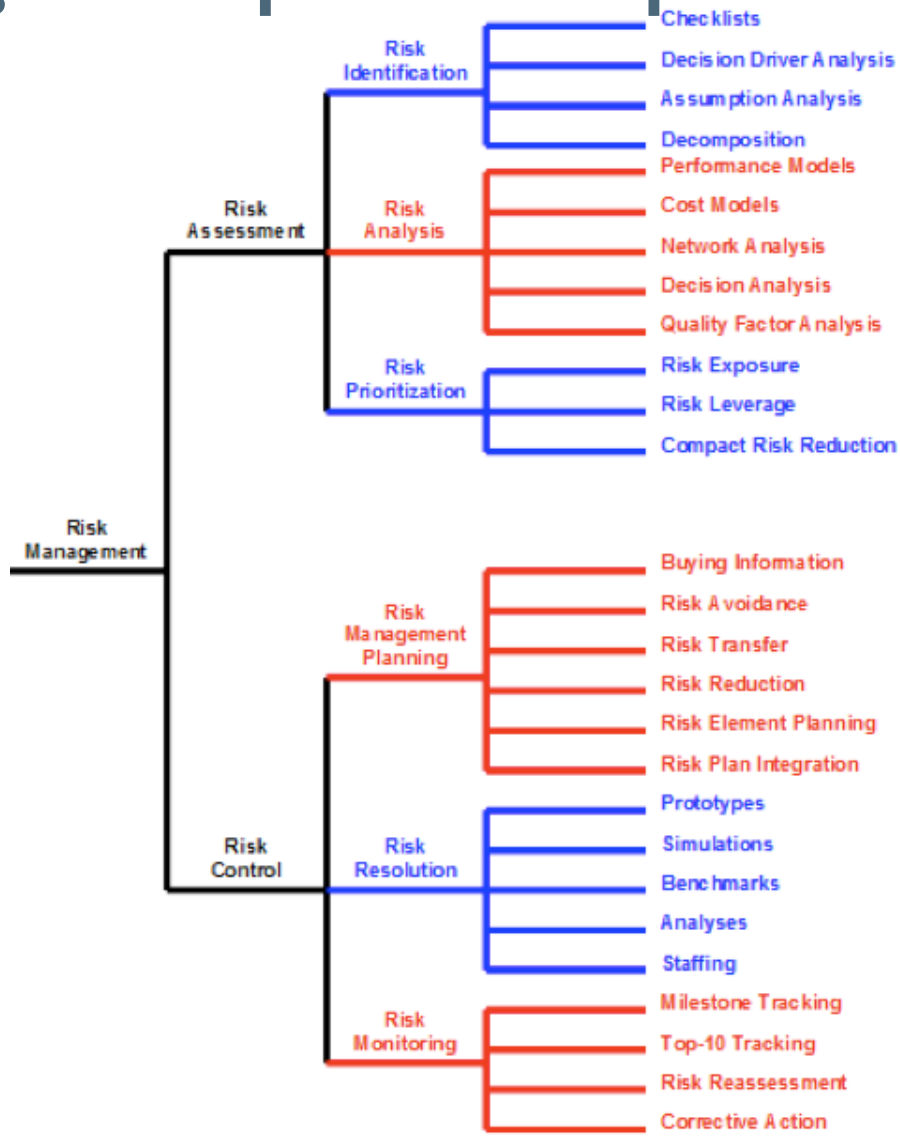
Risk management is highly dependent on individuals.

Known risks typically are not managed.

Top 3 risks are resources, requirements, and the customer-supplier interface.

Pressman, 2005

Risk management process steps



Boehm, 1991

XP and Risk Management

XP's View on Risk

Schedule slips

- The software is not ready at the expected date

Project canceled

- After numerous slips the project is canceled

System goes sour

- After a couple of years the cost of making changes or the defect rate rises so much that the system must be replaced

Defect rate

- The production system is not used since the defect rate is too high

XP's solution

Schedule slips

- Short cycles and frequent releases.

Project canceled

- The Customer chooses the smallest release that give maximal value. So less could go wrong and the value of the software is greatest.

System goes sour

- A comprehensive suite of tests are run after every change. The system is kept in prime condition.

Defect rate

- Tests from perspective of both programmers and customer.

XP's View on Risk II

Business misunderstood

- The software solves the wrong problem

Business changes

- The business problem the software is designed to solve is replaced

False feature rich

- The software includes a lot of features which were fun to program, but fail to make the customer much money

Staff turnover

- Programmers start to hate the program and leave

XP's solution II

Business misunderstood

- The customer is an integral part of the team.
- The specifications are continuously refined.

Business changes

- Shorter release cycles
- During a release the customer could substitute or provide new functionality

False features

- Only the highest priority tasks are addressed

Staff turnover

- Programmers accept responsibility for estimating and completing their own work
- Human contacts are encouraged among the team
- Team members are treated as intelligent species

Reactive Risk Management

Project team reacts to risks when they occur

Mitigation—plan for additional resources in anticipation of fire fighting

Fix on failure—resource are found and applied when the risk strikes

Crisis management—failure does not respond to applied resources and project is in jeopardy

Proactive Risk Management

Formal risk analysis is performed
Organization corrects the root causes of risk

- TQM concepts and statistical SQA
- Examining risk sources that lie beyond the bounds of the software
- Developing the skill to manage change

Risk Management

Identify top 10 risk items
Plan for resolution of each item
Update risk items and plan monthly
Highlight risk item status in progress reviews
Initiate corrective actions

Pressman, 2005

Building a Risk Table

Risk	Probability	Impact	RMMM
			Risk Mitigation Monitoring & Management

Risk Exposure (Impact)

The overall risk exposure, RE, is determined using the following relationship [HAL98]:

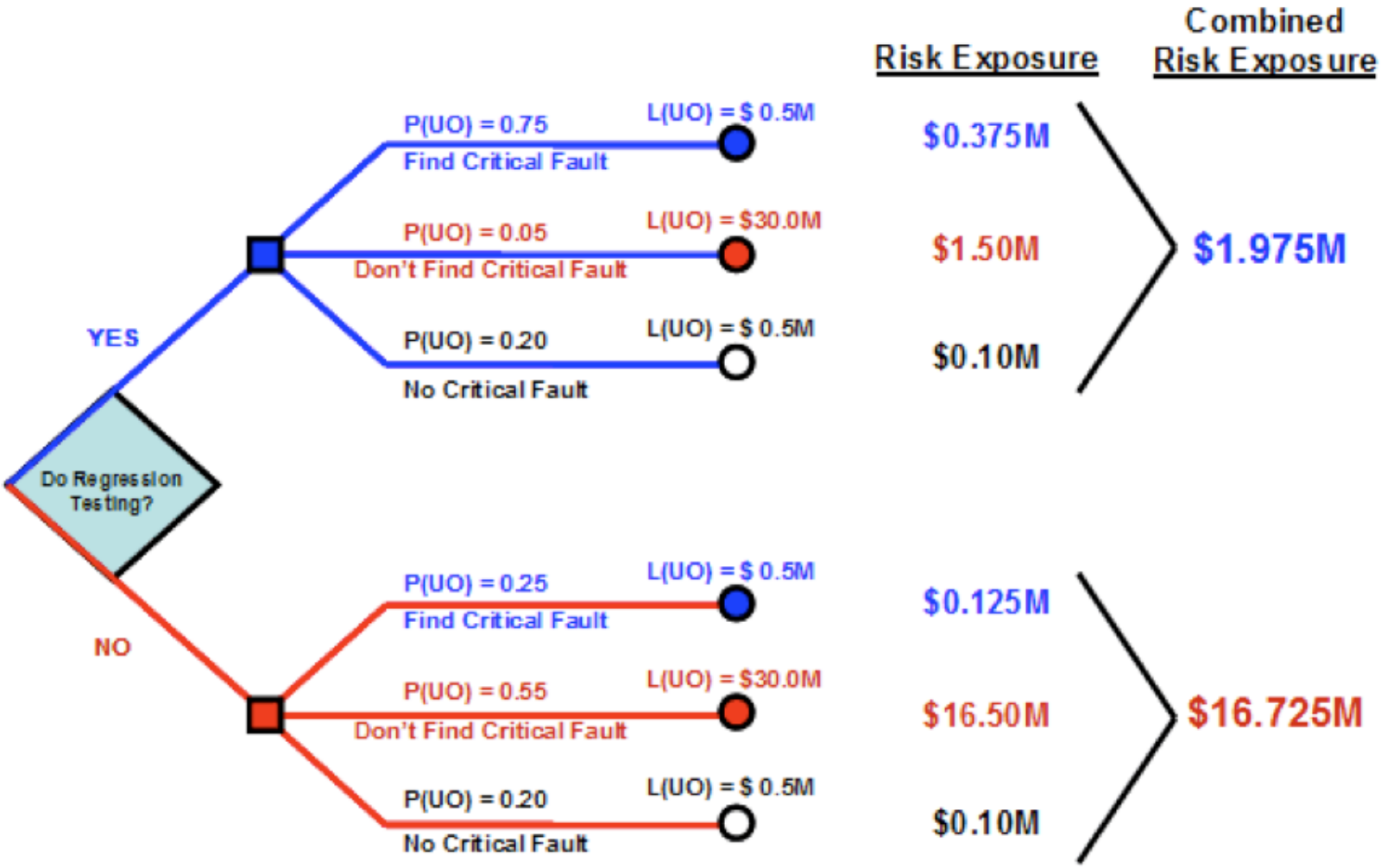
- RE = P(UO)*L(UO)

where

P(UO): probability of unexpected outcome, and

L(UO): loss to the parties affected if outcome is unsatisfactory.

Risk analysis and prioritization



Building the Risk Table

Estimate the probability of occurrence
Estimate the impact on the project on a scale of 1 to 5, where
1 = low impact on project success
5 = catastrophic impact on project success
sort the table by probability and impact

Pressman, 2005

Recording Risk Information

Project: Embedded software for XYZ system
Risk type: schedule risk
Priority (1 low ... 5 critical): 4
Risk factor: Project completion will depend on tests which require hardware component under development. Hardware component delivery may be delayed
Probability: 60 %
Impact: Project completion will be delayed for each day that hardware is unavailable for use in software testing
Monitoring approach:
Scheduled milestone reviews with hardware group
Contingency plan:
Modification of testing strategy to accommodate delay using software simulation
Estimated resources: 6 additional person months beginning 7-1-96

Balancing Methods

Using Risk to Balance Agile and Plan-Driven Methods

Agile and plan-driven home grounds

Project characteristics	Agile home ground	Plan-driven home ground
Application		
Primary goals	Rapid value, responding to change	Predictability, stability, high assurance
Size	Smaller teams and projects	Larger teams and projects
Environment	Turbulent, high change, project focused	Stable, low change, project and organization focused
Management		
Customer relations	Dedicated onsite customers, focused on prioritized increments	As-needed customer interactions, focused on contract provisions
Planning and control	Internalized plans, qualitative control	Documented plans, quantitative control
Communications	Tacit interpersonal knowledge	Explicit documented knowledge

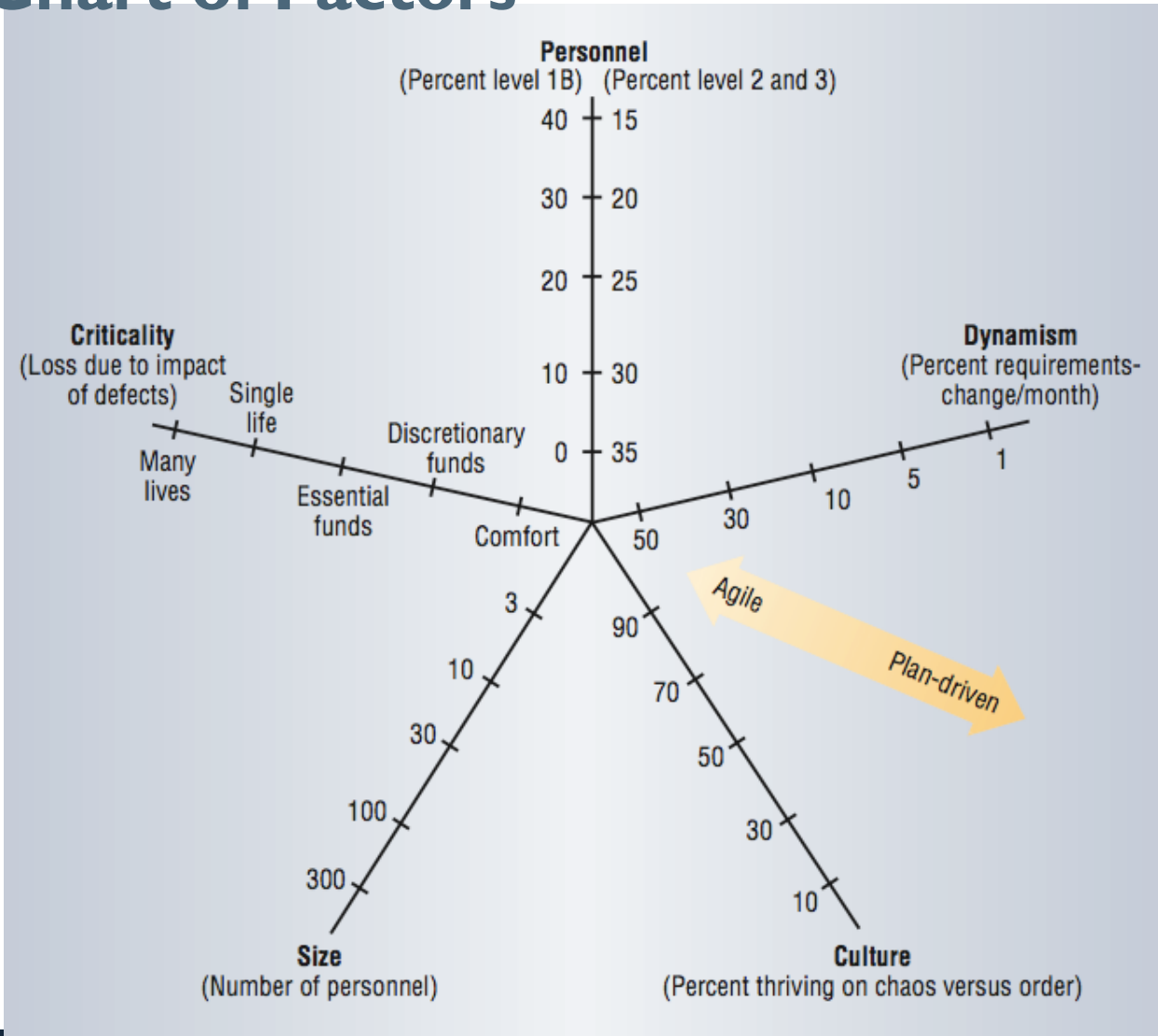
Agile and plan-driven home grounds

Project characteristics	Agile home ground	Plan-driven home ground
Technical		
Requirements	Prioritized informal stories and test cases, undergoing unforeseeable change	Formalized project, capability, interface, quality, foreseeable evolution requirements
Development	Simple design, short increments, refactoring assumed inexpensive	Extensive design, longer increments, refactoring assumed expensive
Test	Executable test cases define requirements, testing	Documented test plans and procedures
Personnel		
Customers	Dedicated, colocated Crack* performers	Crack* performers, not always colocated
Developers	At least 30% full-time Cockburn Level 2 and 3 experts; no Level 1B or Level –1 personnel**	50% Cockburn Level 3s early; 10% throughout; 30% Level 1B's workable; no Level –1s**
Culture	Comfort and empowerment via many degrees of freedom (thriving on chaos)	Comfort and empowerment via framework of policies and procedures (thriving on order)

Critical Factors

Factor	Agility discriminators	Plan-driven discriminators
Size	Well matched to small products and teams; reliance on tacit knowledge limits scalability.	Methods evolved to handle large products and teams; hard to tailor down to small projects.
Criticality	Untested on safety-critical products; potential difficulties with simple design and lack of documentation.	Methods evolved to handle highly critical products; hard to tailor down efficiently to low-criticality products.
Dynamism	Simple design and continuous refactoring are excellent for highly dynamic environments, but present a source of potentially expensive rework for highly stable environments.	Detailed plans and "big design up front" excellent for highly stable environment, but a source of expensive rework for highly dynamic environments.
Personnel	Require continuous presence of a critical mass of scarce Cockburn Level 2 or 3 experts; risky to use nonagile Level 1B people.	Need a critical mass of scarce Cockburn Level 2 and 3 experts during project definition, but can work with fewer later in the project—unless the environment is highly dynamic. Can usually accommodate some Level 1B people.
Culture	Thrive in a culture where people feel comfortable and empowered by having many degrees of freedom; thrive on chaos.	Thrive in a culture where people feel comfortable and empowered by having their roles defined by clear policies and procedures; thrive on order.

Polar Chart of Factors



Method Determination