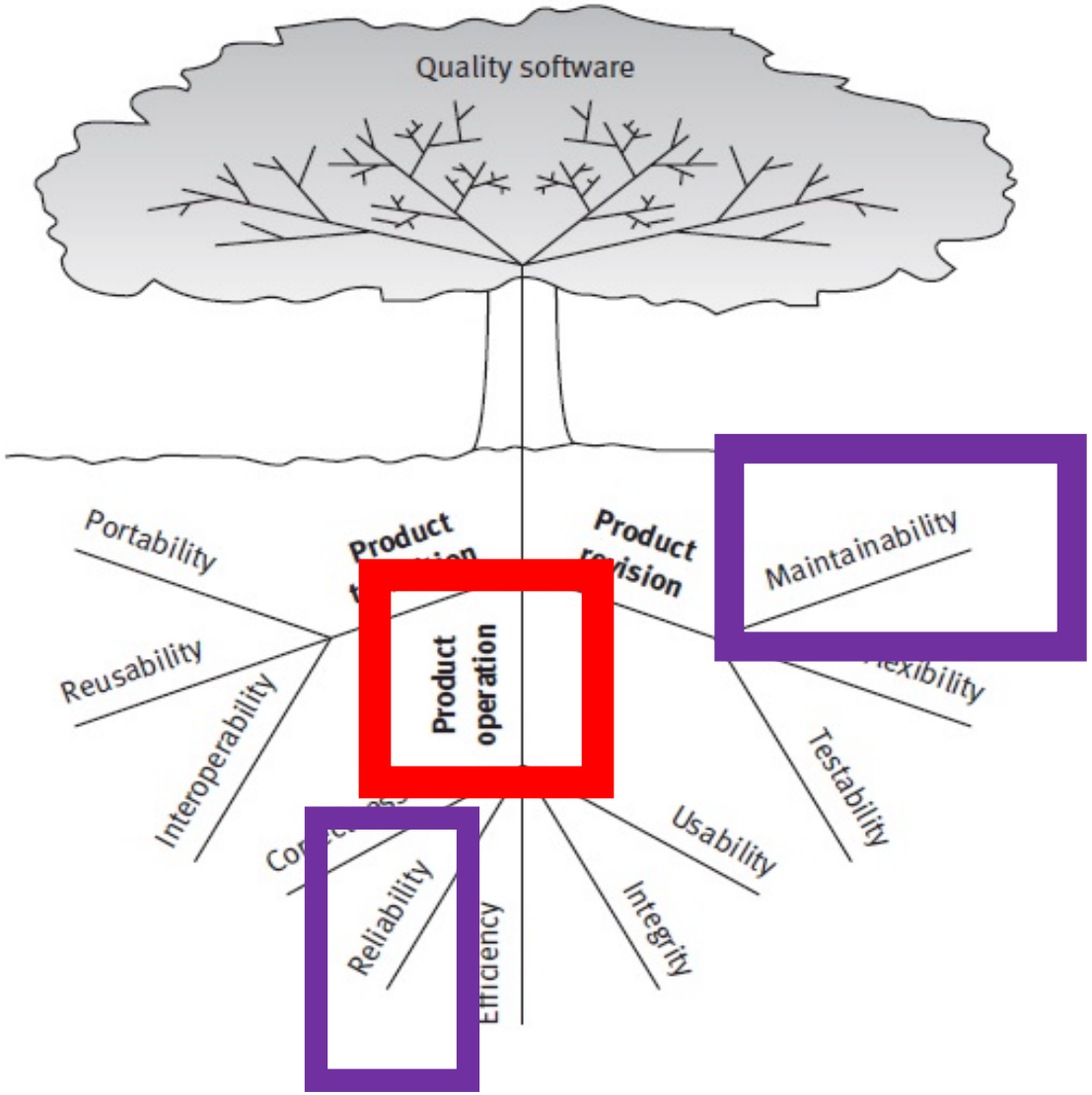
Course 5

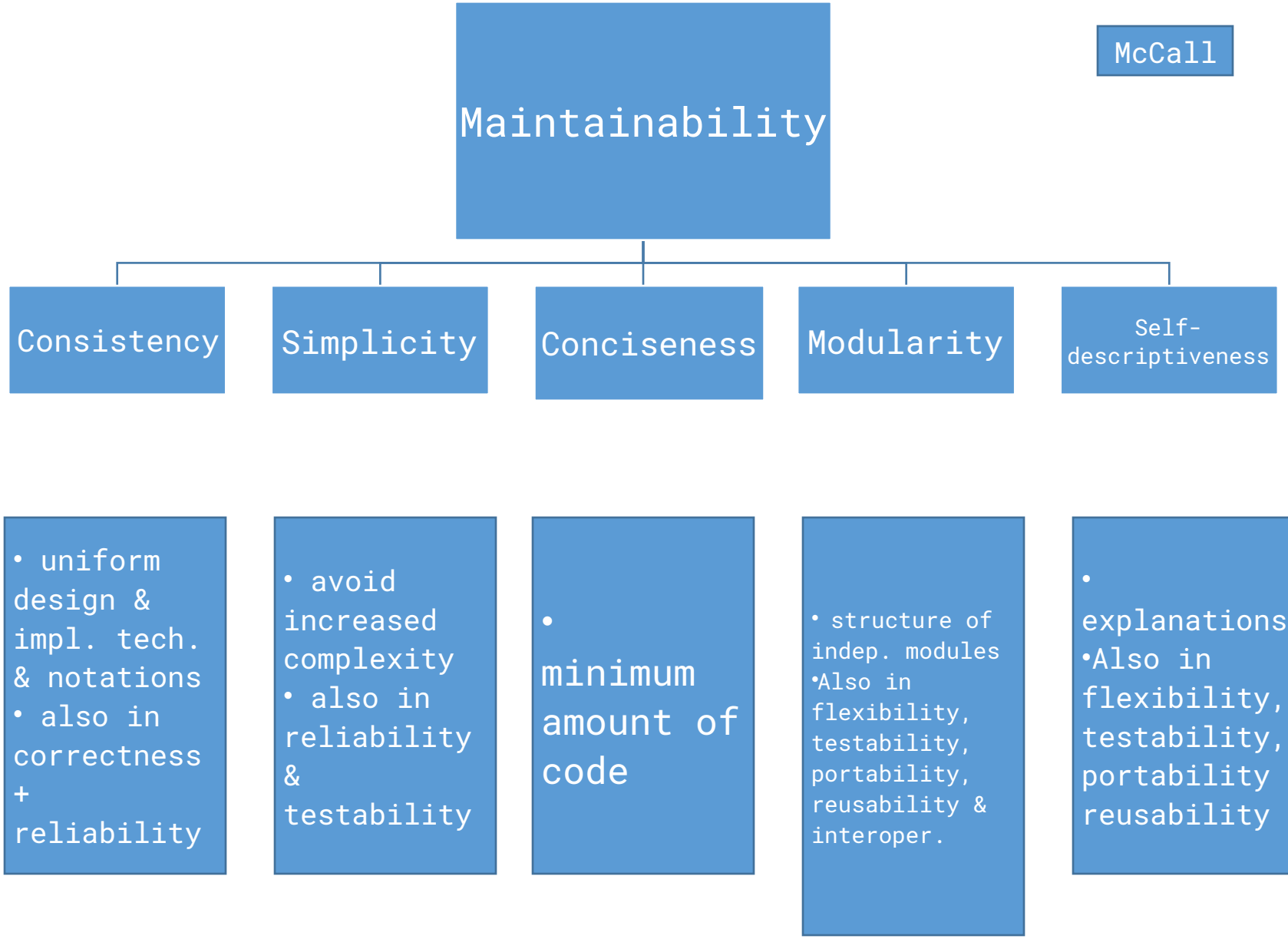
Maintainability, Reliability


Quality software
Portability
Reusability
Interoperability
Product operation
Maintainability
Testability
Usability
Integrity
Efficiency
Reliability

Maintainability [McCall]

- Definition: Effort required to locate and fix an error in an operational program

- Impact:
  - Measured: design + implementation
  - Realized in: maintenance + transition


Maintainability
McCall
Consistency
Simplicity
Conciseness
Modularity
Self-descriptiveness
• uniform design & impl. tech. & notations
• also in correctness + reliability
• avoid increased complexity
• also in reliability & testability
• minimum amount of code
• structure of indep. modules
•Also in flexibility, testability, portability, reusability & interoper.
• explanations
•Also in flexibility, testability, portability & reusability

Maintainability measured …

- Maintainability index

```
MI = 171-5.2 x ln(aveV)
       -0.23 x aveV(g')
       -16.2 x ln(aveLOC)
       + 50 x sin sqrt(2.4 PerCM)
http://www.virtualmachinery.com/sidebar4.htm
[-100, 200]
```

Maintainability measured …

- Maintainability index

```
MI = MAX(0,(171 - 5.2 * ln(Halstead Volume) –
       0.23 * (Cyclomatic Complexity) –
       16.2 * ln(Lines of Code))*100 / 171)
[0,100]
0-9 =bad; 10-19= satisfactory;  20-100 = acceptable
```

Parameter	Name	Measures
aveV	Average Halstead complexity	Computational density
aveV (g')	Average extended cyclomatic complexity	Logical complexity
aveLOC	Average count of lines of code	Code size
PerCM	Average percent of lines of comments	Human insight

Halstead complexity:

$$V = N \times \log_2(n)$$

N = no of operators

n = no of distinct operators

Cyclomatic complexity:
= number of linearly independent paths through a program

Maintainability influenced by…

- Analyzability
- Changeability
- Stability – encapsulation and data hiding
- Testing
- Development environment features

Maintainability influenced by…

➢ Easily identify elements to change
➢ Changes compared to the specification
➢ Changes affect rest of the system

➢ Locate causes of failure correct
➢ Locate parts for modification – extend

➢ Readability
➢ Comprehensibility
➢ Traceability

➢ Analysis: browsing & searching
➢ Build: automated
➢Testing
➢ Commit: privileges

➢ Unit testing
➢ Integration testing
➢ System testing
➢ Test coverage

Advice:
• style: name convention, indent
• simple !!!
• comments

Dependencies & coupling

- Data coupling
- Stamp coupling:
  - Large data structure is passed, and only a part is required
- Control coupling
- Common coupling – global variables
- External coupling: diff. modules sharing common resources
- Content coupling: module modifies or relies on another module

Maintainability measured …

- 3 purposes:

  System maintainability over time – disintegrate as it evolves

  Compare different systems performing the same task

  Evaluate parts – less maintainable – target for refactoring

Maintainability measured …

- Tools:
  - Lachesis Eclipse plug-in
    http://lachesis.sourceforge.net/
  - JetBrains's IntelliJMetricsReloaded plug-in
    http://www.jetbrains.com/idea/
    - Metrics – .NET framework

Maintainability for OOP

- WMC – Weighted Methods per Class
- DIT – Depth of Inheritance Tree
- NOC – Number of Children
- CBO – coupling between objects (method call, field access, inheritance, exceptions, …)
- …

Reliability

Reliability

- Definition
  - ➢ in engineering: the ability of a product or component to continue to perform its intended role over a period of time to pre-defined conditions
  - ➢software systems: is measured in terms of the mean time between failures, the mean time to repair, the mean time to recover, the probability of failure and the general availability of the system.
- Aspects:
  - Maturity – absence of software faults that lead to failures
  - Robustness (Fault tolerance) – performance to specification despite some faults
  - Recoverability – operation after a failure

Reliability issues

- Input problems
- Output problems
- Logic problems
- Computation problems
- Data-handling problems
- Fault tolerance

Input problems

- DON’T accept wrong input (out-of range values)
- Accept all possible correct input

- Reject unreasonable input – careful
- Consider missing input elements

Output problems

- Incomplete or missing output

Adding an element to a program and forgetting to update output routines

- Correct results at the wrong time

Sensitive to time, especially for files, printer, …

- Wrong format

- Internal representation vs. human readable representation
- special output displays

Logic problems

- Associated with program's control flow
- Affect operations and data
- Difficult to detect and locate
- Typical:
  - Off-by-one errors
  - Neglecting extreme conditions
  - Forgotten cases, condition tests or steps
  - Misinterpretation of specification

- Missing processing or counting the first or last element of a range

Values outside range

Computation problems

- Incorrect algorithm or computation
  - Example 1:
    - Evaluate an expression using Polish Postfix Form
    - Input: 7-4-3; output: 6
  - Example 2:
    - Evaluate a or b and c
- Uninitialized variables
- Referencing NULL
- Abusing type system

Data-handling problems

- Incorrect data initialization
- Referencing wrong data variable
- Out-of-bounds values
- Inconsistent data – representation – add, update, delete

Fault tolerance

- Fault = software or hardware problem in a component of a system under examination\
- Fault – leads to a system failure: behavior deviated from specification requirement
- Fault-tolerance system:
  - detect and diagnose the fault
  - report/ mask/ compensate it

  => Avoid failure

Reliability
McCall model

- Definition: extend to which a program can be expected to perform its intended function with required precision
- Includes:
  - Availability
  - Accuracy
  - Robustness
  - Precision
  - Tolerance

Key factor in applications:

- Human lives affected
- Very high system development cost
- Real-time

Criteria

- Error tolerance
- Consistency
- Accuracy
- Simplicity

Measured in:

- Analysis
- Design
- Implementation & debugging

Impact on:

- Testing
- Operation (execution)
- Maintenance

Evaluation (see McCall.pdf)

- Based on criteria:
  1. Consistency: procedure + data consistency
  2. Accuracy
  3. Error tolerance: error tolerance, input data, recoverable computational failures, recoverable hardware faults, device status condition
  4. Simplicity: design structure, programming paradigm (~~structured~~), data and control flow complexity, code simplicity

Tools for Reliability Evaluation

- Isograph – Reliability Workbench: Fault Tree Analysis, Reliability Block Diagram Analysis, Event Tree Analysis, Markov Analysis a.s.o.
- Item – ToolKit prediction and analysis suite: Markov module, FaultTree, ARALIA
- Relex – Relex prediction and analysis suite: 17 module for reliability prediction, data analysis a.s.o.

Explanation of terms

- Fault Tree Analysis – identifies and ranks combinations of events that can lead to system failure
- Markov analysis – calculates reliability based on simulation
- ARALIA – quick decision processing, based on statistical distribution of failure probabilities, computes exact measure of event importance, …

Fault Tree Analysis (FTA)

- deductive failure analysis based on Boolean logic

> Deductive ⇒ backwardly deduce the causes of the event

- 1962 – Bell Laboratories – aviation
- Used in reliability and safety analysis
- Used in SE – debugging

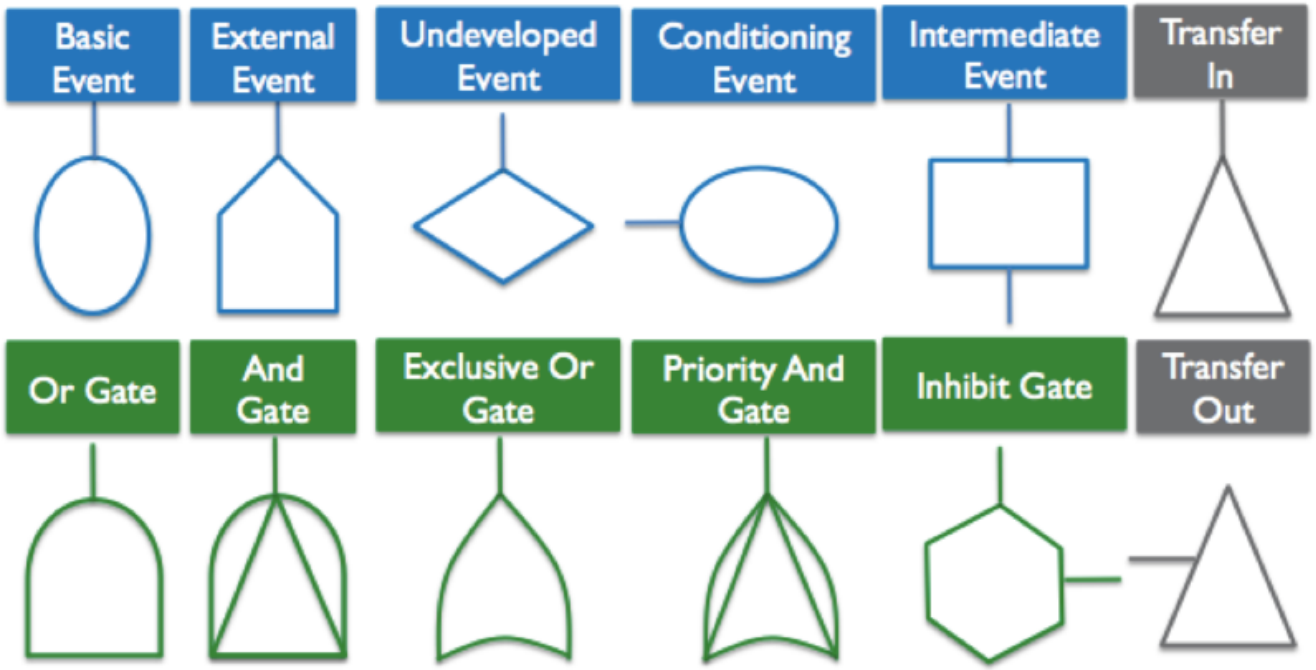
Fault Tree Analysis (FTA)

- Analyze an undesired state of the system – combine series of lower-level events:

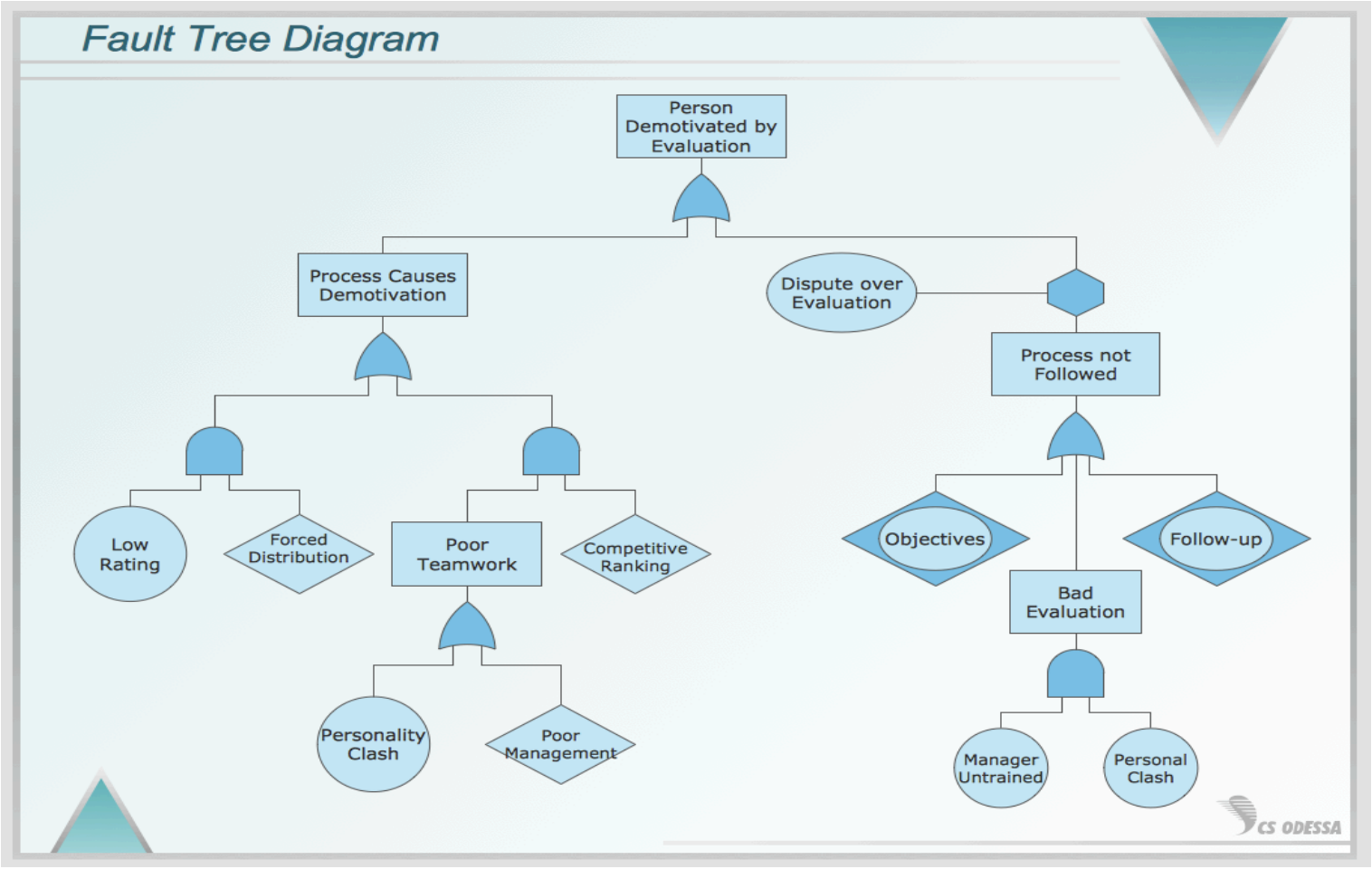
1) Defined undesired event
2) The event is resolved into its immediate causes
3) This resolution of events continues until basic causes are identified
4) A logical diagram called a fault tree is constructed showing the logical event relationships

Fault Tree Analysis (FTA)

- Graphical symbols:
  - Event
  - Gate
  - Transfer



FTA- example


Fault Tree Diagram
Person Demotivated by Evaluation
Process Causes Demotivation
Dispute over Evaluation
Process not Followed
Low Rating
Forced Distribution
Poor Teamwork
Competitive Ranking
Objectives
Follow-up
Bad Evaluation
Personality Clash
Poor Management
Manager Untrained
Personal Clash
CS ODESSA

try

- http://www.fault-tree-analysis-software.com