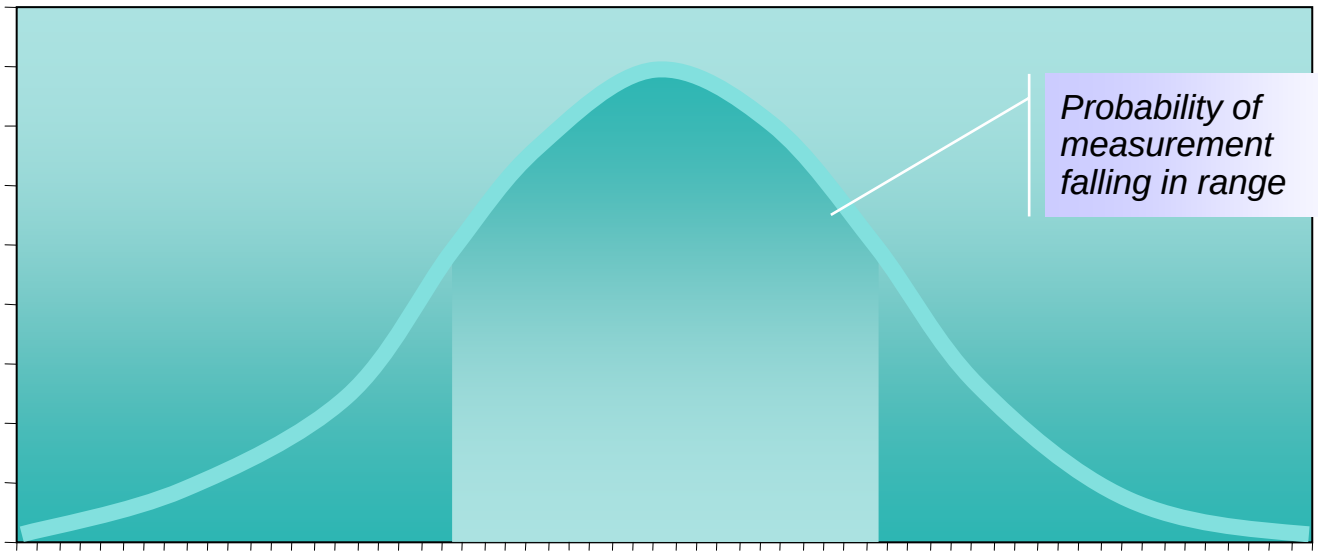
IBM Software Group

Practical Approaches to Development Governance

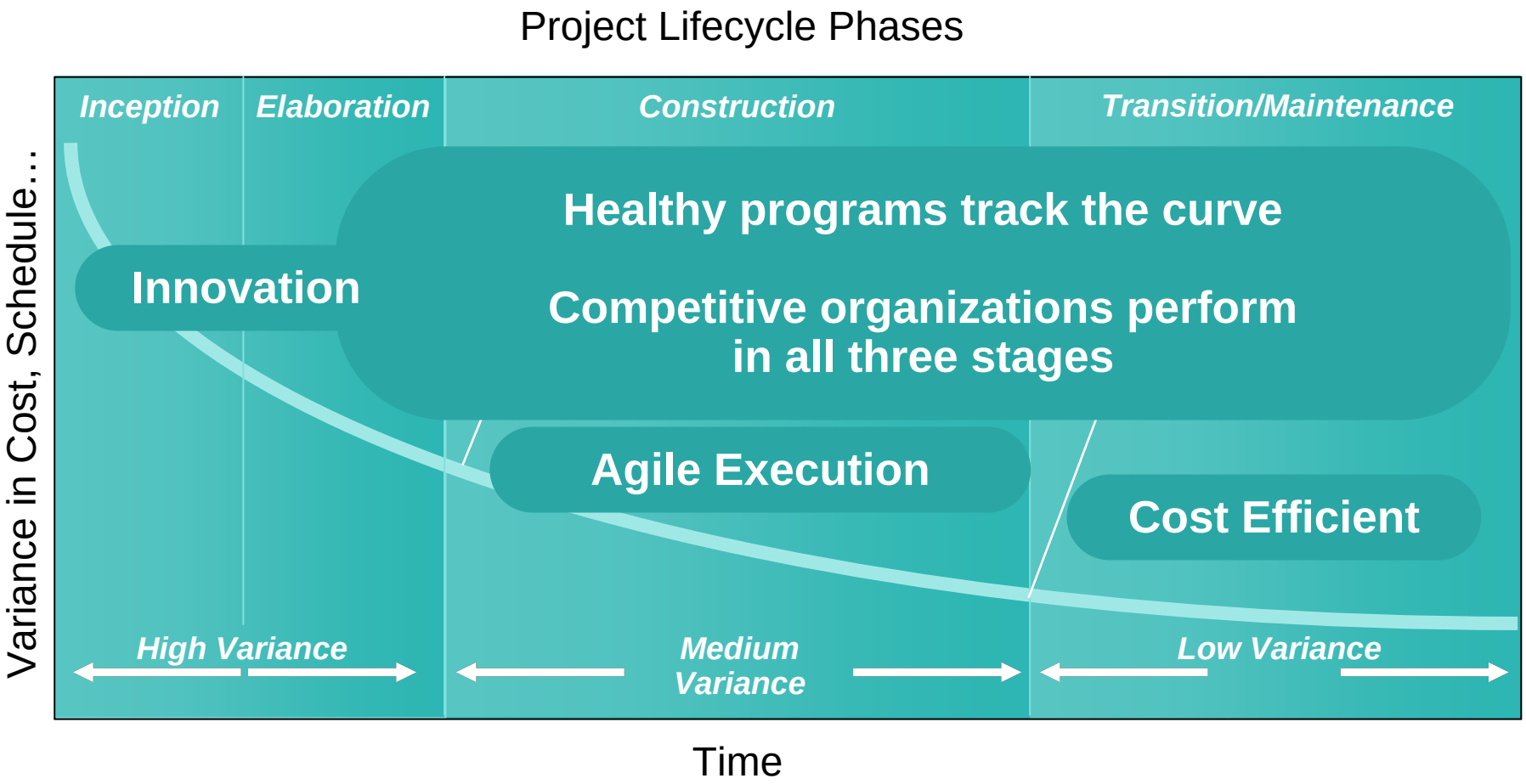
Rational. software

Go to IBM

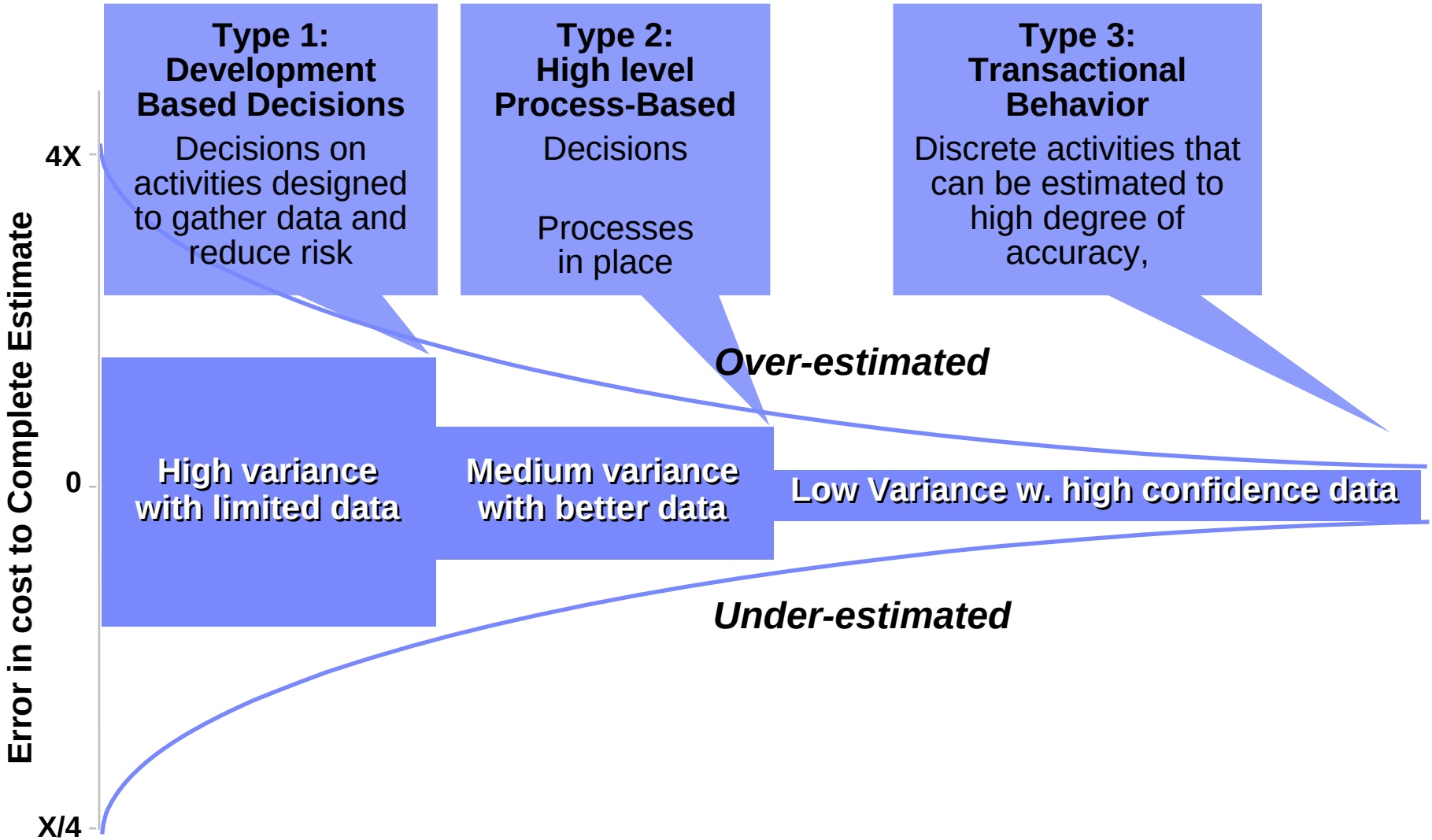
Program parameters (cost, schedule, effort, quality, …) are random variables



Development Dynamics



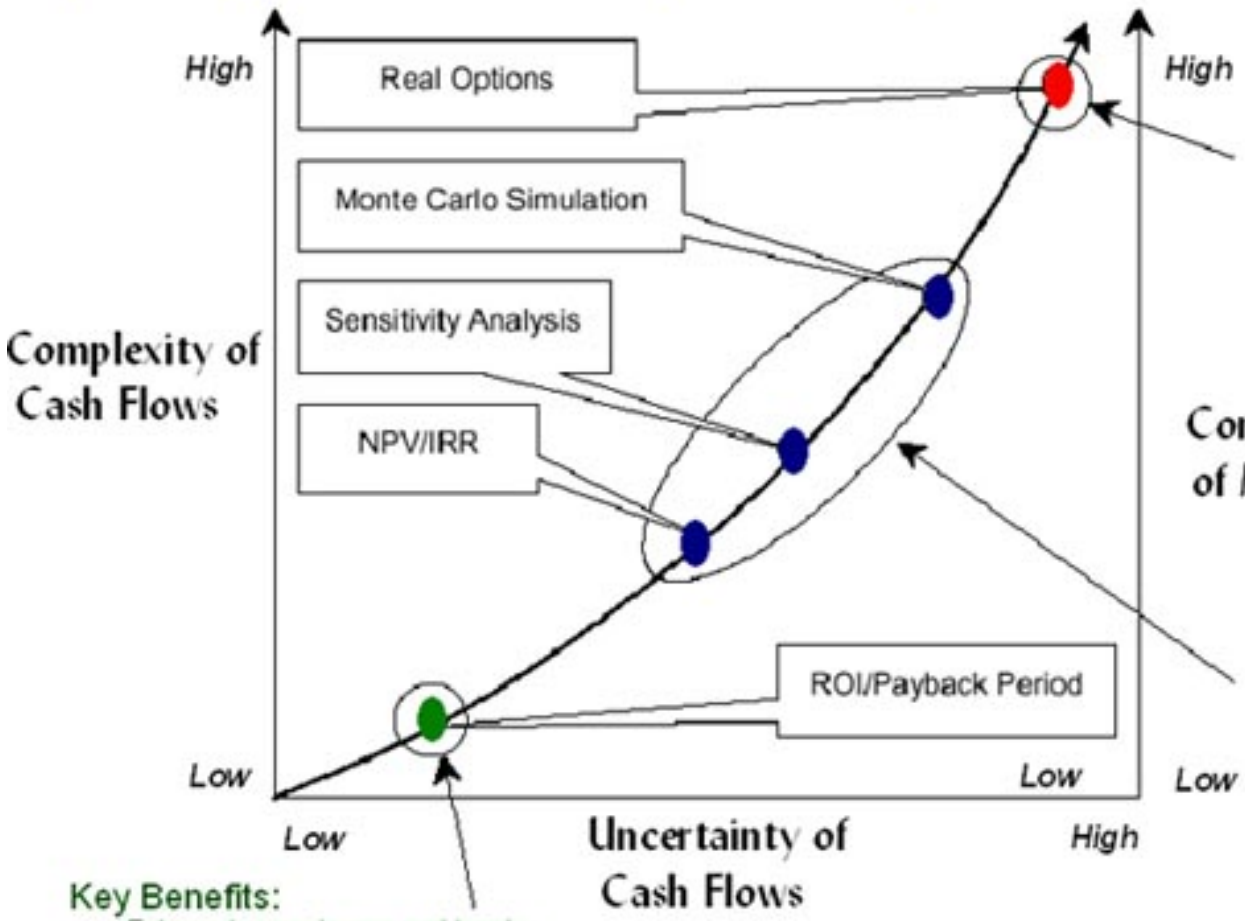
Process Phases Should Focus on Risk Reduction



Project Cost Comments

- **Organizations spend time at phase I**
  - Gathering data to make decisions
  - Not all parameters are known
  - Defining requirements
  - Defining architecture
- **Organizations spend money at phase II and III**
  - Test and Product Support get added to development costs
  - Development gets defect fixing overhead in addition to new development
  - Test adds more tests as defects are uncovered in product usage
  - In addition, more platforms are supported and the regression tests become larger
  - Product support becomes larger as product usage becomes pervasive

Applicability of Valuation Analysis Techniques



Key Benefits
- Highly flexible and accurate

Applicability:
- Particularly well suited to technology investments; extreme level of complexity may limit applicability to only large, intricate investments

Key Benefits
- Able to value cash flows temporally and adjusted for risk

Applicability:
- Well suited to all categories; Monte Carlo simulation and sensitivity analysis particularly well suited to technology and capability investments

Key Benefits:
- Extremely easy to use and teach

Applicability:
- Best suited to short-term utility investments; both metrics lose utility when project is risky, complex, long-lived or competing with alternative investment opportunities

The Key To Remember Is That Different Valuation Techniques are Required for Different Levels of Risk

- One of the great weaknesses of NPV is that it assumes business conditions will remain constant for the duration of the analysis
- This is rarely the case in software unless the duration is on the order of a year or at most two
- Thus to account for the risk of change, one has to use things like real options or sensitivity analysis
- Similarly in changing conditions of risk in the software development lifecycle, one has to use metrics and techniques that adapt to the changes in risk over the lifecycle
- However, there are some areas of the lifecycle that are amenable to standard techniques.

Metrics Recommendation

ON DEMAND BUSINESS™

A Few Rules For Measurement

- Trends are More Important than Point Values
- Initially use metrics to improve projects rather than compare projects
- Initially defect density used to measure risk. In later stages defect density used to measure product quality
- Forecasts initially used to measure risk, at the end used to plan the projects

Strategy for Metrics

- In Initial phases with high development risk,
  - Objective is revenue growth
  - Measure a few parameters very often
    - Low overhead, also low latency for data
    - In addition much volatility of metrics so they need to be tracked often
    - This will allow for adjustments to be made quickly in the projects
    - Metrics should cover the entire lifecycle
    - Initial phases should be characterized by fast cycle times
- As revenue risk increases, more testing and support required thus cycle time increases
  - Processes becomes more predictable
  - Objective becomes cost efficiency
  - More metrics can be taken to improve process
  - More metrics have greater granularity for cost improvement

If You Are Not Measuring Any Metrics Start With These:

- Defect Density – Measures Quality of product
- Requirements Volatility – Measures Quality of Requirements and churn
- Project Management Metrics – Cost Performance Indicators (CPI) and Schedule Performance Indicators (SPI)
- Source Lines of Code for the Project
  - As time goes on source lines of code per service
  - One can also start looking at Function Points

Defect Management Evolution

- High Risk – Type 1
  - Defect Density
  - Defects By Priority
  - Defects by Age
- Medium Risk- Type 2
  - Defect Density
  - Defects By Priority
  - Defects by Age
  - Defects by Service
- Low Risk – Type 3
  - Defect Density
  - Defects By Priority
  - Defects by Age
  - Defects by Service

Requirements Management Evolution

- High Risk – Type 1
  - Use Cases:
    - Analyzed
    - Traced to Tests
    - Volatility
  - **URPS Requirements**
    - Analyzed
    - Traced to Tests
    - Volatility
- Medium Risk- Type 2
  - Use Cases:
    - Analyzed
    - Designed
    - Traced to Tests
    - Volatility
  - **URPS Requirements**
    - Analyzed
    - Traced to Use Cases and Tests
    - Volatility
- Low Risk – Type 3
  - Use Cases:
    - Analyzed
    - Designed
    - Traced to Tests
    - Volatility
  - **URPS Requirements**
    - Analyzed
    - Traced to Use Cases
    - Traced to Tests
    - Volatility

Validation and Test

- High Risk – Type 1
  - Test Cases Planned
  - Test cases Failed
  - Total test cases executed
- Medium Risk- Type 2
  - Test Cases Planned
  - Test cases Failed
  - Total test cases
  - Test Coverage
  - Test case Volatility
  - URPS Metrics
  - URPS Margins
- Low Risk – Type 3
- All of above plus
  - Test Case Volatility by service
  - Test Cases Failed by service

Iterative Development – Project Management

- High Risk – Type 1
  - Schedule and Cost performance indices
- Medium Risk- Type 2
  - Predicted Size
  - Predicted Productivity
  - Actual Productivity
- Low Risk – Type 3
  - Predicted Defect Density
  - Predicted Defects Per Unit Time

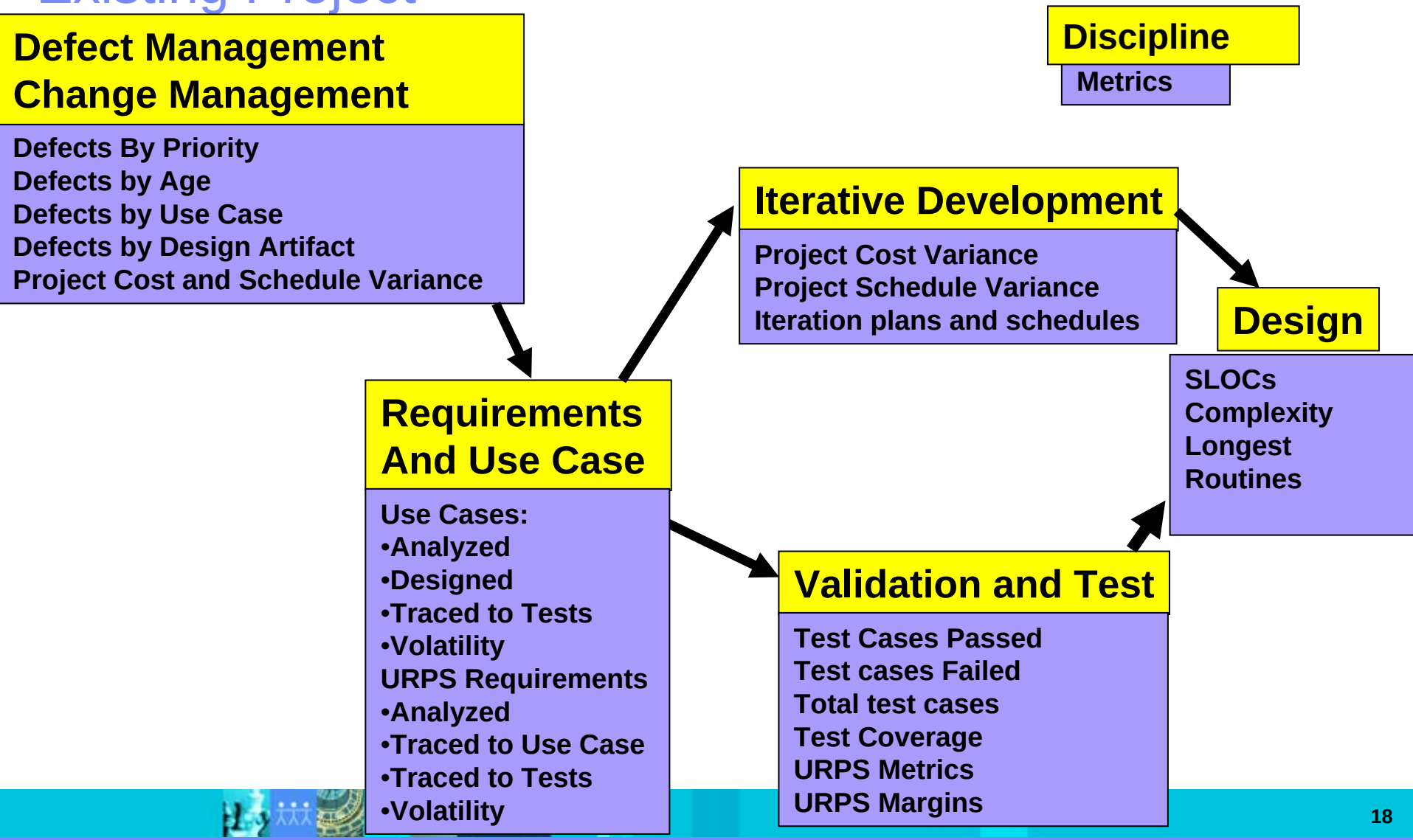
Design Metrics

- High Risk – Type 1
  - Total SLOCs or Function Points
- Medium Risk- Type 2
  - Total SLOCs or Function Points
  - Complexity
  - Pattern Scan Results
  - Longest routines
- Low Risk – Type 3
  - Total SLOCs or Function Points
  - Complexity
  - Pattern Scan Results

Deployment Thoughts

- Start with the back end of the process- Defect Management and Build
  - Allows for measurements and baselines changes
  - Reduces risk of process changes causing defects at customer ship
  - Fastest return
- Then go to the requirements management process
  - Bounds SDLC
  - Starts to mitigate highest risk of SDLC
- Very soon after work on testing and iterative development
  - Testing is highest cost
  - Iterative Development is a generally a major culture change
  - Finish with design

Ordering of Implementation For End to End on an Existing Project



The Facts are Our Friends

- When metrics show problems, support the reporter
  - Sometimes the news may not be good
  - Do not shoot the messenger
  - No better way to stop getting the facts
- Use the metrics to support the strategic goals of the organization