Executive Risks

Common Risks that Executives Are Unaware Of

Risk Categorization

- Efficiency
- Effectiveness

Specific Risks

- **Effectiveness**
  - Requirements
  - Technical
  - Architecture

- **Efficiency**
  - Process
  - Executive
  - Schedule
  - Cost
  - Time

Many software projects fail because we try to manage effectiveness by focusing on efficiency

Accelerated Development

Efficiency vs. Effectiveness

- **Effectiveness**
  - Are you building the correct system?
- **Efficiency**
  - Are you building it with the least time, cost, and resources?

Being efficient is pointless if you are wasting energy building the wrong software system

Accelerated Development

Requirements Risk

- **Incomplete requirements**
  - Requirements required for core architecture are very costly to add late in a project

- **Inconsistent requirements**
  - Leads to the team having a different view of what the system does
  - This leads to continual fire fighting

Incomplete or inconsistent requirements virtually guarantees that your project will be overtime/over budget or cancelled.

Technical Risk

- **Implementation technology**
  - Inability to implement key requirements
  - Insufficient for actual performance requirements

Failing to identify key performance up front can lead to discovering technical short falls late in the project

Architecture Risk

- **Insufficient skills**
  - Architect does not have sufficient skills
- **Incorrect core architecture**
  - Missing requirements lead to missing core architecture

Core architecture is built in the first months
Missing key architecture can be difficult/impossible to add later

Process Risk

- **Inter-department handoffs**
- **Insufficient Development Infrastructure**
  - Version control
  - Issue tracking
- **Bug triages**

Insufficient processes will bring production releases to a halt

Executive Risk

- **Non-technical Executives**
  - Unawareness of catastrophic risks to techical projects
- **Infeasible projects**
  - Executives decreeing functionality, time, and resources

Core architecture is built in the first months
Missing key architecture can be difficult/impossible to add later

Common Project Risks

- Incomplete and insufficient requirements
- Incorrectly chosen implementation technology
- Poor core architecture
- Insufficient process for coordination
- Poor bug triages
- Uneducated executives

Non-software companies often have 3 or more of these issues which leads to late or cancelled projects

Requirements Architecture Interaction
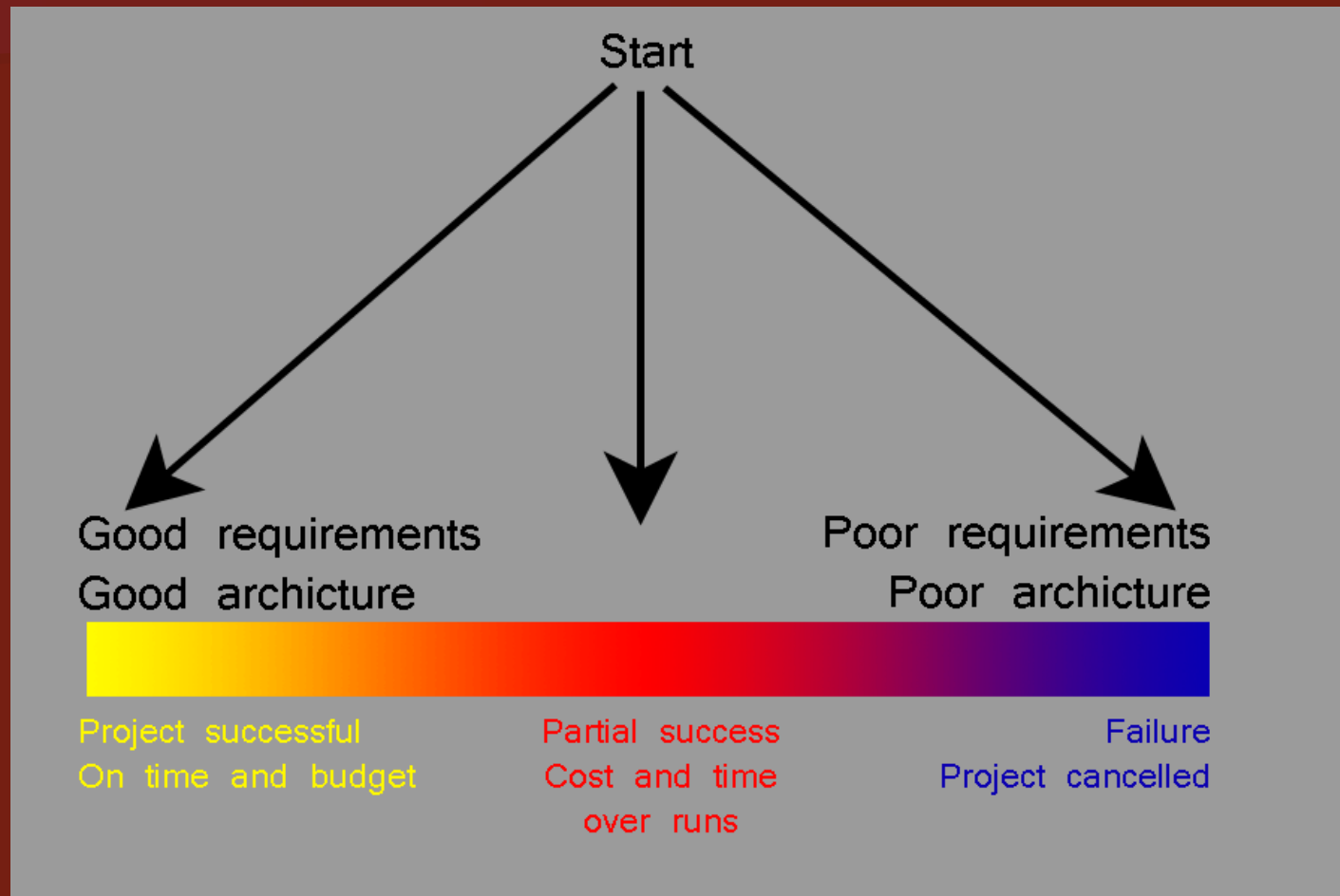

Good requirements

Mediocre requirements

Poor requirements

Good architecture | Mediocre architecture | Poor architecture
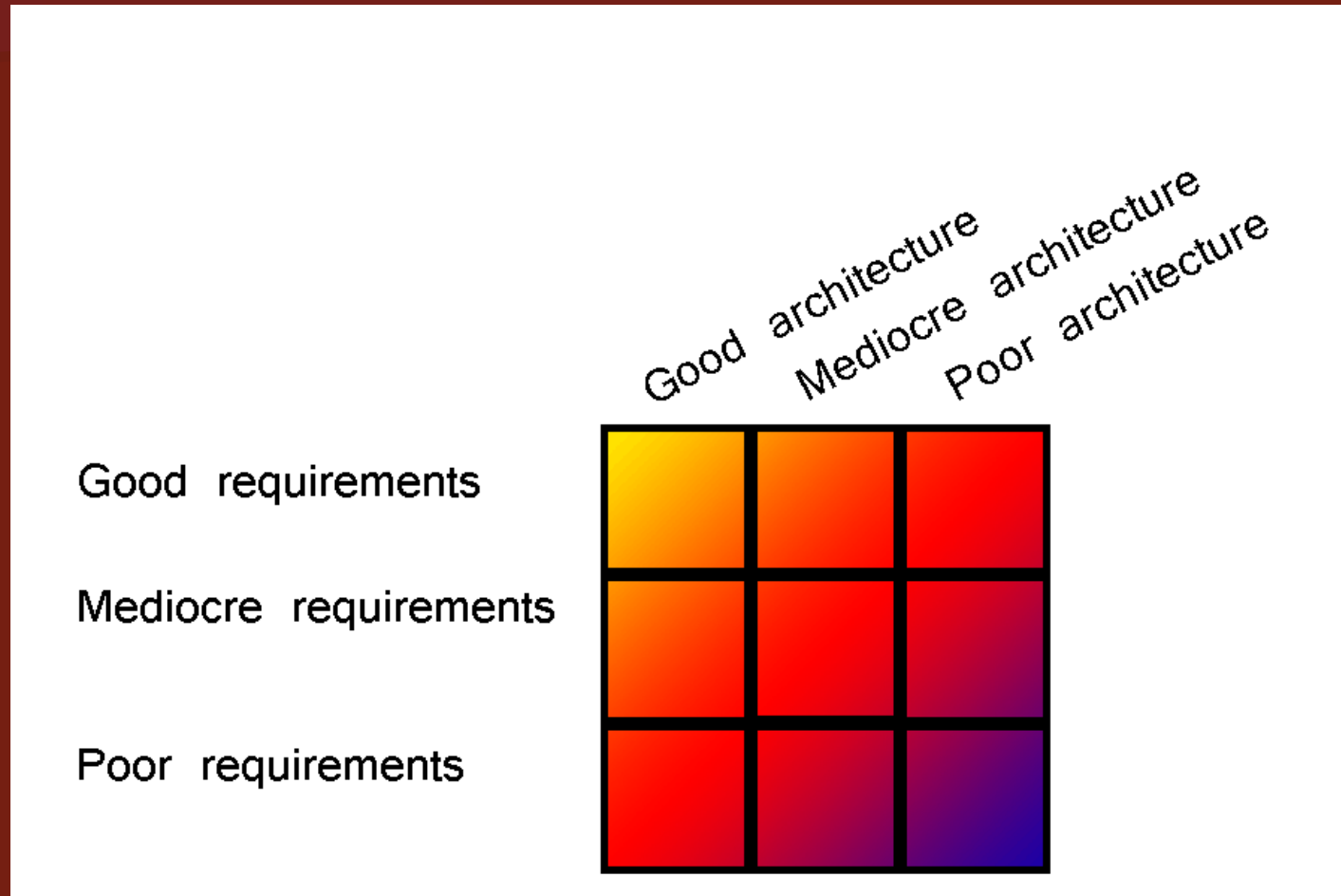
Requirements Architecture Interaction

Accelerated Development

Summary

■ **Inflexible software**
  - occurs when the effectiveness and efficiency risks are not addressed
  - occurs within the first few months of development

Inflexible software is discovered around the 1st usable release
After the initial few months it is too late to fix it

Accelerated Development

Conclusion

- **To prevent inflexible software**
  - All efficiency and effectiveness risks must be dealt with BEFORE the project starts
  - The architect for the system must have skills to select technology and understand the technical requirements
  - Projects need to address core requirements and architecture very early in a project