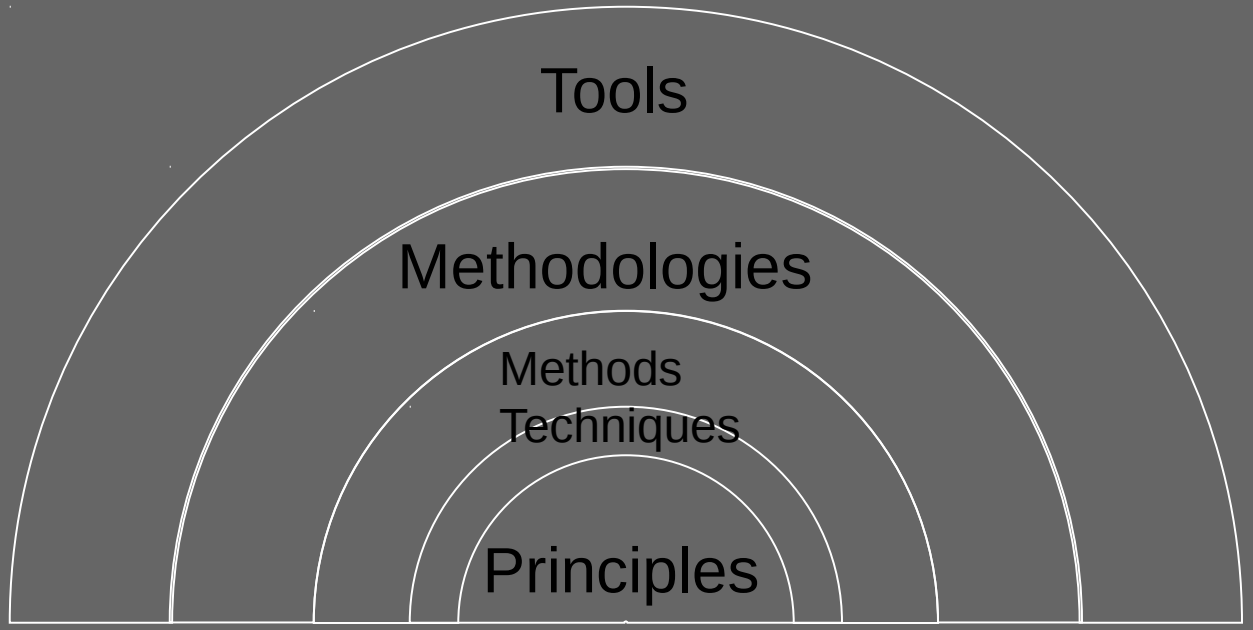
Fundamental Ideas

Seven Principles of Software Engineering

Overview

- Software Engineering is based on a collection of fundamental principles
- These principles guide the development of all aspects of software development
  - Languages
  - Methods
  - Tools
  - Process
  - Project Management

How it All Relates



We Will Cover

- Rigor and Formality
- Separation of Concerns
- Modularity
- Abstraction
- Anticipation of Change
- Generality (?)
- Incrementality

Rigor and Formality

- Software development is a creative process
  - Creativity implies informality and chaos
- Rigor and formality seems to contradict creativity

- Not necessarily so
  - Increase the confidence in the creative results

- Evident in
  - Programming languages, design notations, requirements specifications, process definitions

Separation of Concerns

- We cannot deal with all aspects of a problem simultaneously
  - One way to conquer complexity
- Separate issues and tasks
  - Separate functionality from efficiency
  - Requirements specification from design
- Various types of separation
  - In terms of time
  - In terms of qualities
  - In terms of views of an artifact

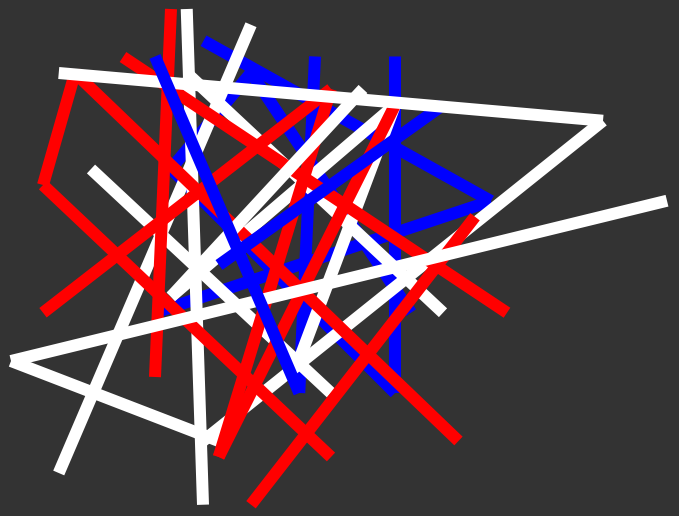
More on Separation

- By separating concerns we may miss out on optimizations
  - True
  - **But, if we take a global view we may fail entirely!**
- Separation of concerns allow separation of responsibilities
  - Separation of managerial and technical issues
  - Separation of requirements and design

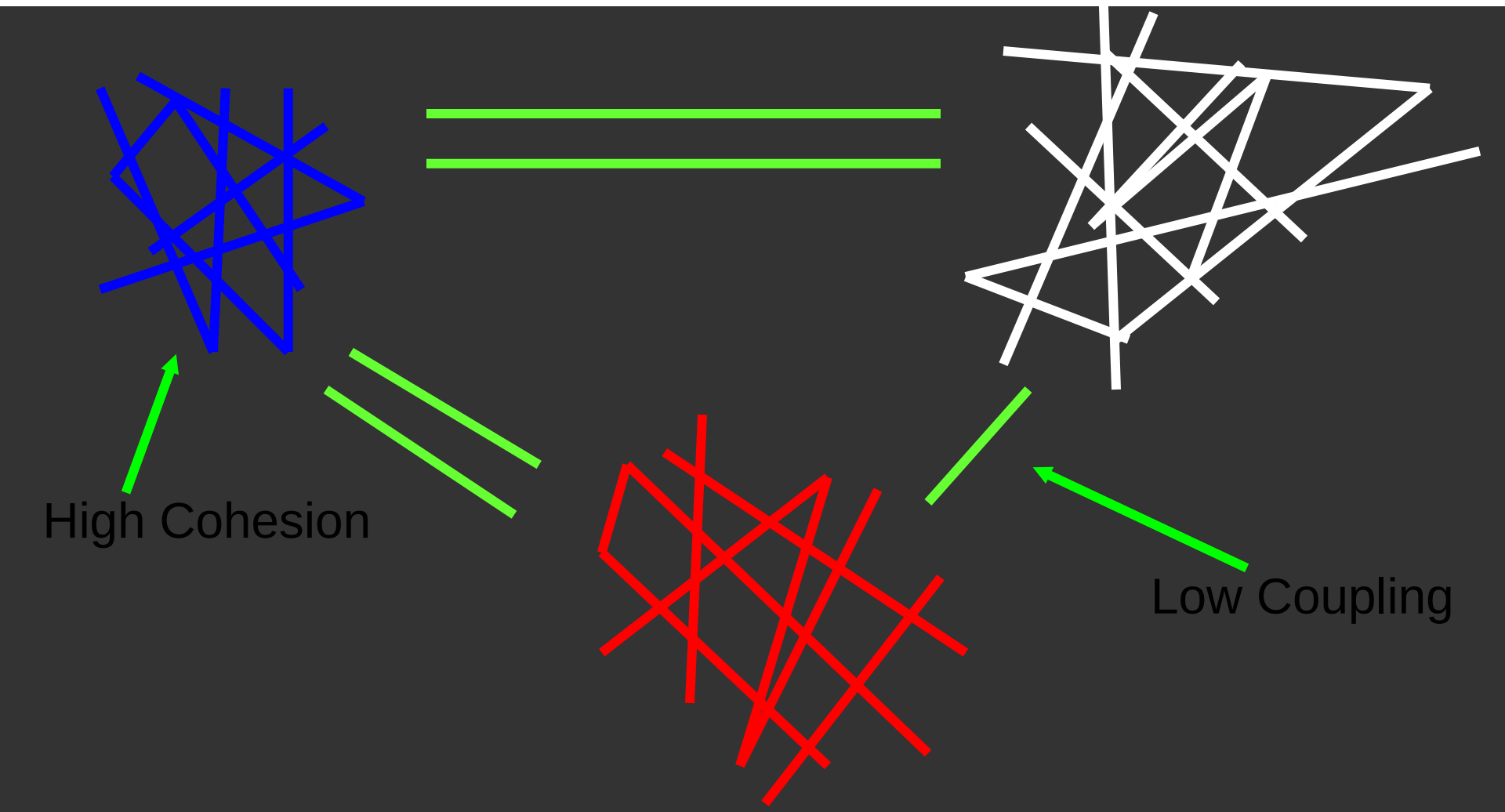
Modularity

- A complex system must be broken down into smaller modules
- Three goals with modularity
  - Decomposability
  - Break the system down into understandable modules
  - Divide and conquer
  - Composability
  - Construct a system from smaller pieces
  - Reuse, ease of maintenance, OO frameworks
  - Ease of understanding
  - The system will be changed; we must understand it
  - Understand in pieces versus understanding the whole

More Modularity



Two Essential Properties



Abstraction

- Identify the important aspects and ignore the details
- Must have different abstractions of the same reality
  - Provide different views
- Examples of abstraction
  - Design notations
  - Project planning
  - Etc.
- Two key concepts
  - Information hiding and data encapsulation

Anticipation of Change

- Change is inevitable
  - We might as well plan on it!
- This effects all aspects of Software Engineering
  - Make sure all artifacts are easy to change
  - Modularization and separation of concerns
  - Make sure you can maintain many versions of all artifacts
  - Configuration control
  - Plan for personnel turnover
  - Plan for a rapidly changing market
  - Plan for rapidly changing technology

Generality (??)

- In every problem, attempt to find a more general solution
  - General problem is often easier to solve
  - A generalized solution may be reusable
  - If you are lucky, you may even be able to buy instead of build
- Attempt of the software industry to parallel hardware and manufacturing

Incrementality

- Move towards the goal in increments
  - Very difficult to have a “big bang” approach to anything
- Areas where we see incrementality
  - Identify useful subsets of an application and deliver in increments
  - Prototyping
  - All software development adds functions a few at the time
  - The whole development process; the spiral model

Concluding Remarks

- These fundamental principled guide all aspects of software development
- Remember
  - Tools, methodologies, and techniques will evolve, the principles remain the same