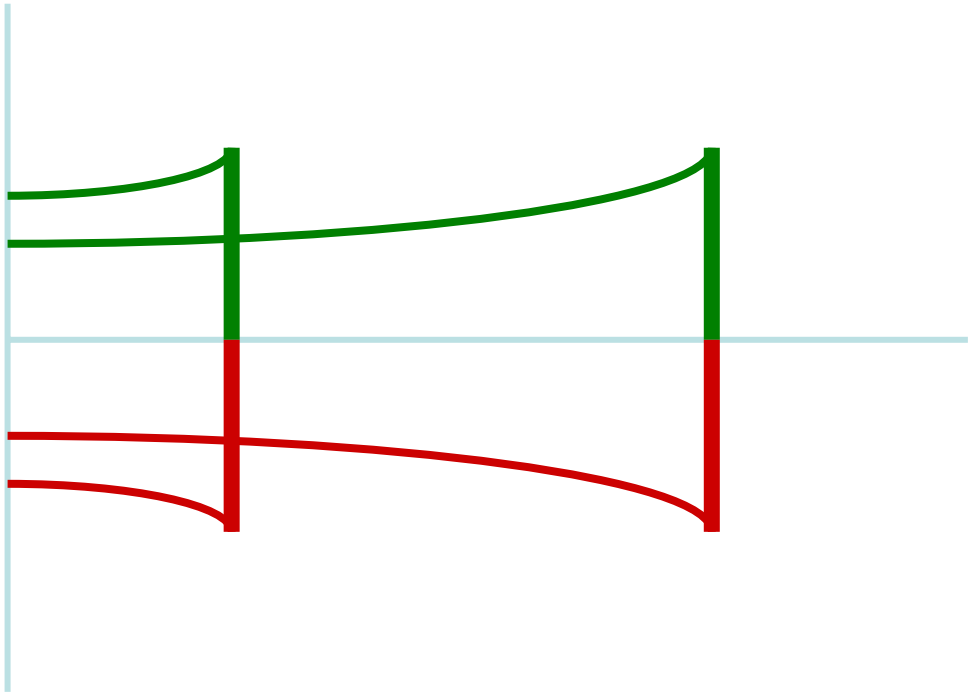
Effective Design

Kent Beck

Three Rivers Institute

Economics

- Time value of money



Unknowns

- Needs
- Means
- Cost
- Usefulness

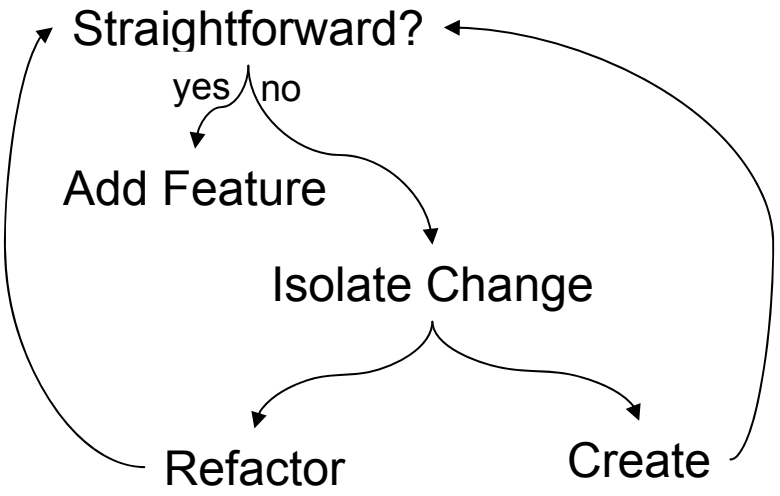
Design is Social

- Experience
- Distribution

Theory

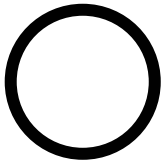
- Beneficially relating elements
- Cost driver: rippling changes
  - Coupling
  - Cohesion
- Scale-free
  - Fractal

Process

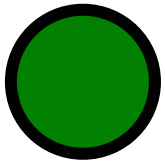

Straightforward?
yes
no
Add Feature
Isolate Change
Refactor
Create

Create

- Principle: safe steps
  - Going back is expensive
- Leap
- Parallel
- Migrate
- Simplify
- Place Stepping Stone

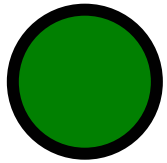


Old design



New design

Leap

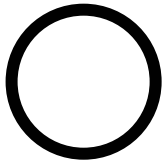


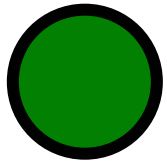
1. Make new design
2. Move all uses
3. Delete old design

\+ Quick

\- Risk of it not working for large changes

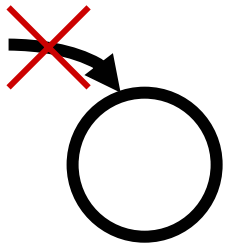
Parallel

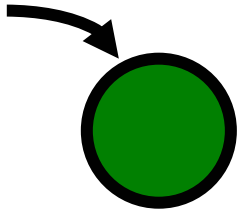




1. Make new design
+ Quick
+ Safe—doesn't disturb existing uses
+ Often used in framework evolution
- Costly to maintain two designs
- Need to figure out how to have them both run

Migrate





1. Move a use

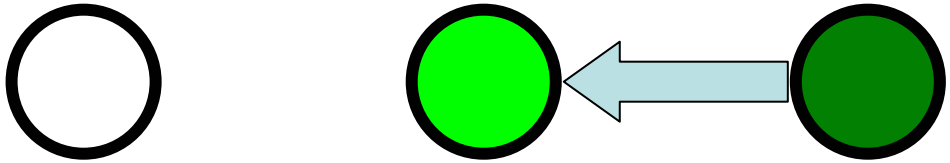
\+ Quick (per migration)

\+ Provides feedback for new design

\+ Low risk

\- Costly to migrate many uses

Simplify



- Eliminate constraints
- Reduce needs
  - One, not many
  - Few, not many
  - Special case, not general

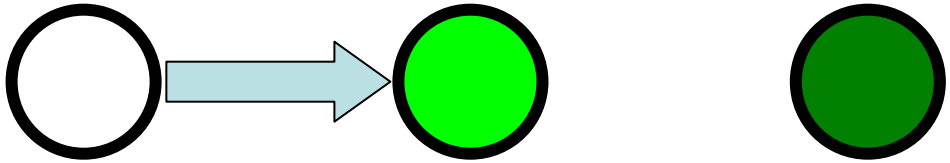
\+ Quick

\+ Safe

\- What if it isn't really progress?

\- What if you ignore the wrong constraint?

Place Stepping Stone



1. Build a language (framework) in which getting to the new design is easier

+ Quicker

- What if it doesn't make the new design easier?
  - Every extra bit is expensive for uses and maintainers

- Responsibility of language designers and implementors is much broader than application developers (build, debug, analyze)

Refactorings

- Isolate changes
- Extract/Inline method/object
- Eliminate/introduce duplication
- Eliminate/introduce abstraction/indirection
  - Interface
  - Superclass
- Move method
- Move field

Conclusion

- Plan backwards from adding straightforward features
- Move in safe steps
- Make progress when you can’t see the end
- If you can’t make progress, add the feature anyway