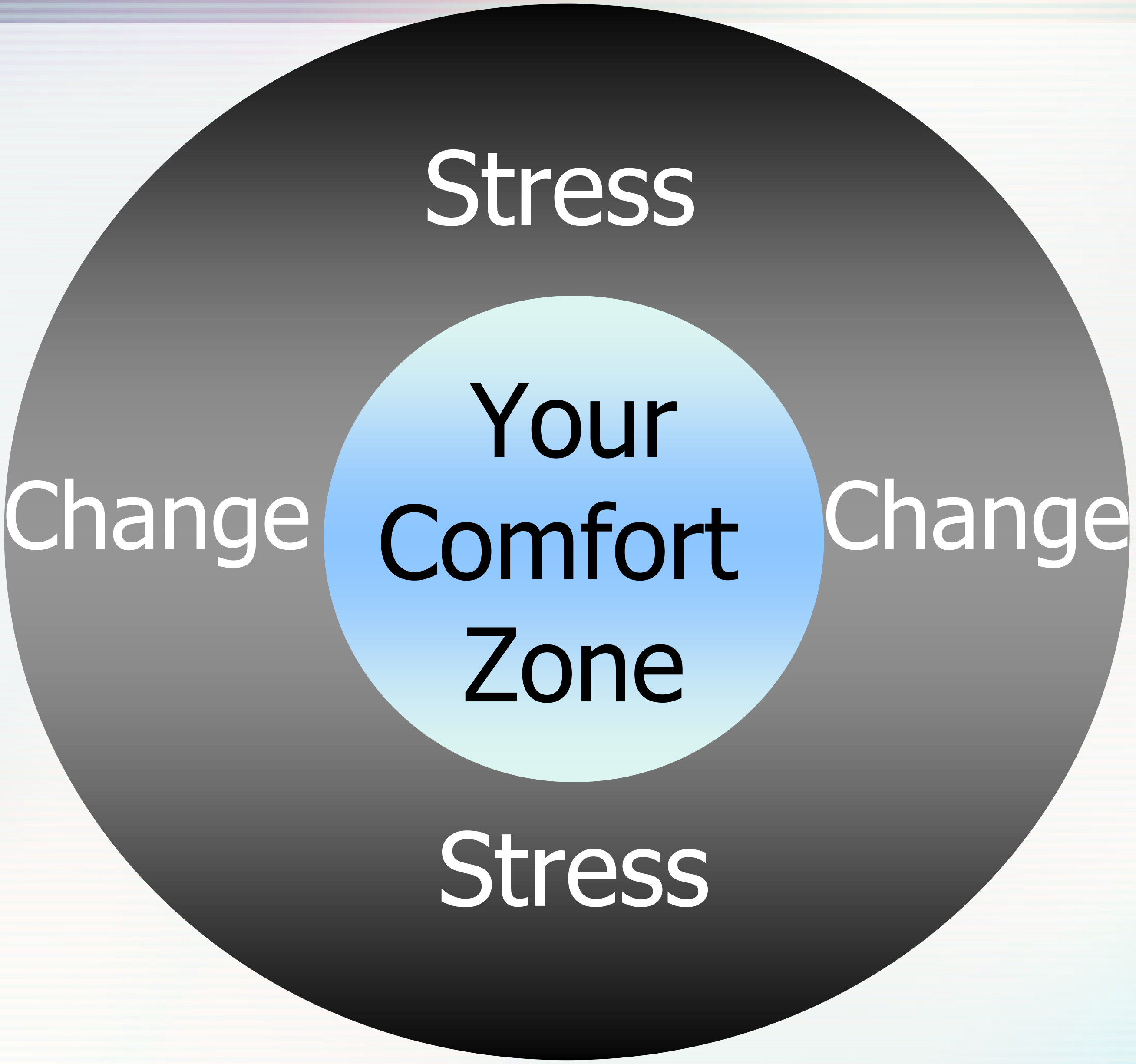
Agile Israel
Feature Driven Development

For the agile agent of change

Justin-Josef Angel

www.JustinAngel.Net

blogs.Microsoft.co.il/blogs/JustinAngel


Stress
Change
Your Comfort Zone
Change
Stress

Agile Israel
Feature Driven Development

For the agile agent of change

Justin-Josef Angel

www.JustinAngel.Net

blogs.Microsoft.co.il/blogs/JustinAngel

Who here has caused change?

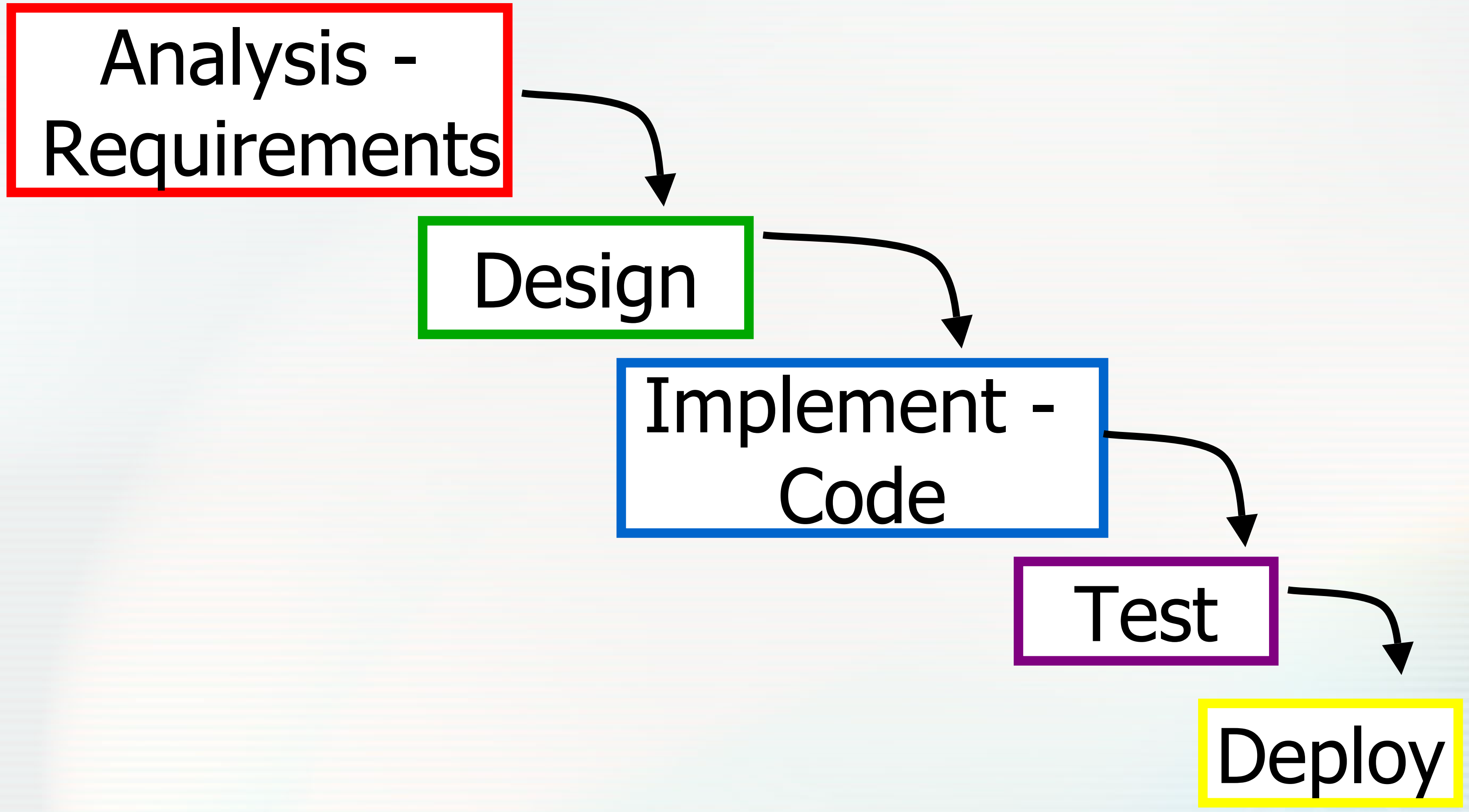
- Deployed an application?
- Upgraded technology?
- Switched IDEs?
- Told someone you loved them?
- People resist to change.
- People don’t want change.
- Change = Bad.

Agile = Change = Bad

- **Agile methodologies require us to work differently.**
- **What about other people?**
- **PROBLEM**
- **But first, What are you going to get from this presentation?**
- **and what is “Agile”?**

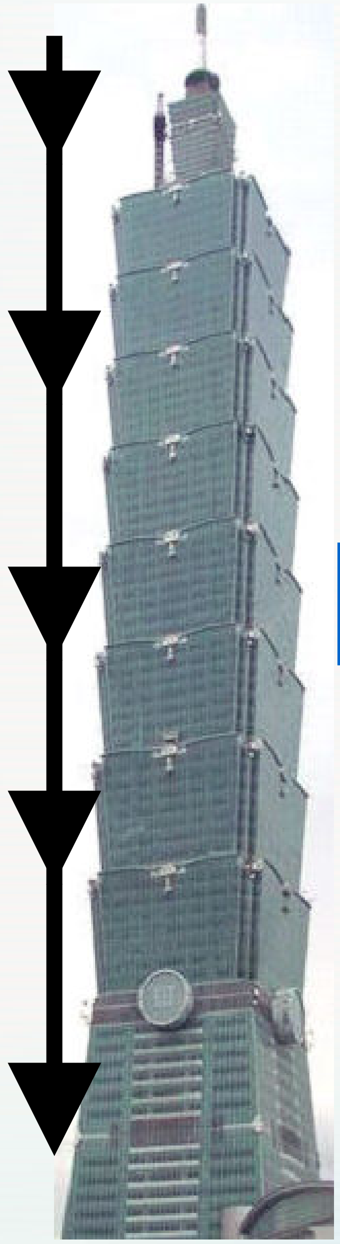
What Is Agile?

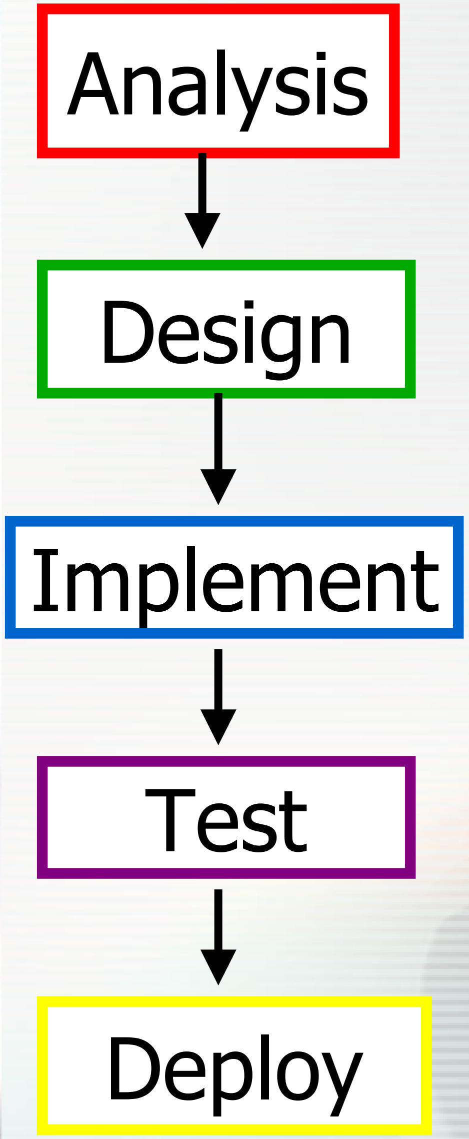
- **Agile is the solution to a problem.**
- **The traditional Waterfall model:**



Waterfall has a problem

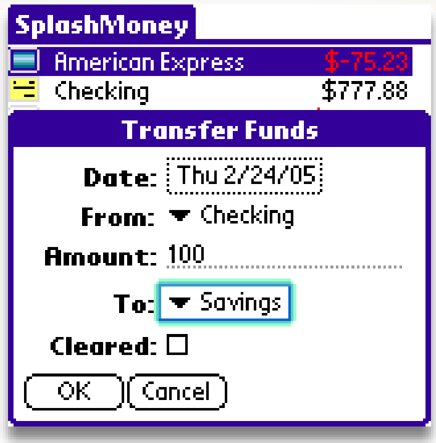
- "building model"
- jumping from the side of a building.
- If you fall from the top – it's really going to hart when you reach the bottom.
- Projects FAIL!





Example of the problem

- "Transfer X amount of money from one account to the other and take 10% commission".
- Who are we taking the commission from?
- Israeli Banks – From the sender.
- Paypal – from the receiver.


SplashMoney

American Express	$-75.23
Checking	$777.88

Transfer Funds

Date: Thu 2/24/05

From: Checking

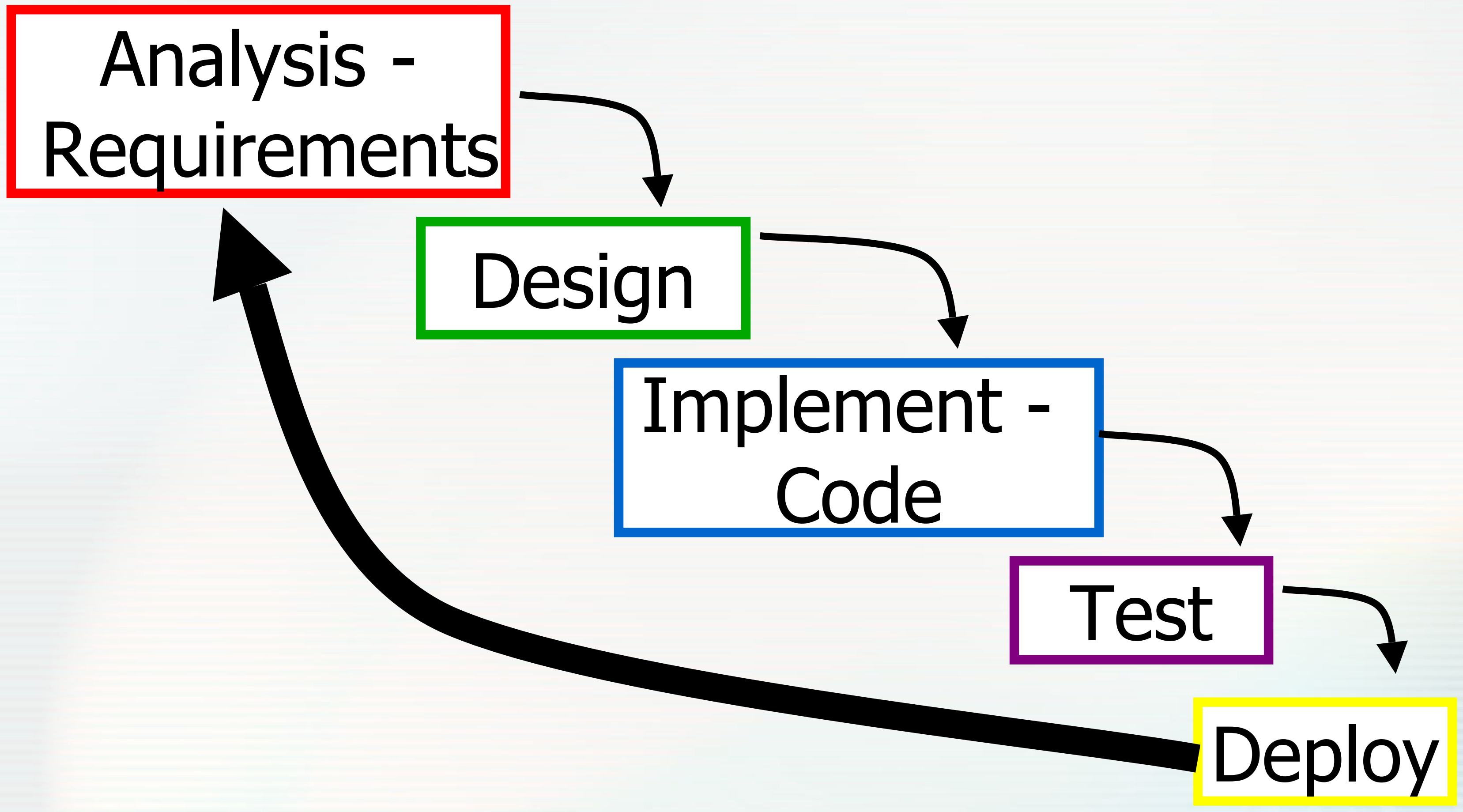
Amount: 100

To: Savings

Cleared: ☐

Simplest Solution – Short Iterations

- **Take the waterfall model – and add one arrow.**



What is Agile and is it the solution?

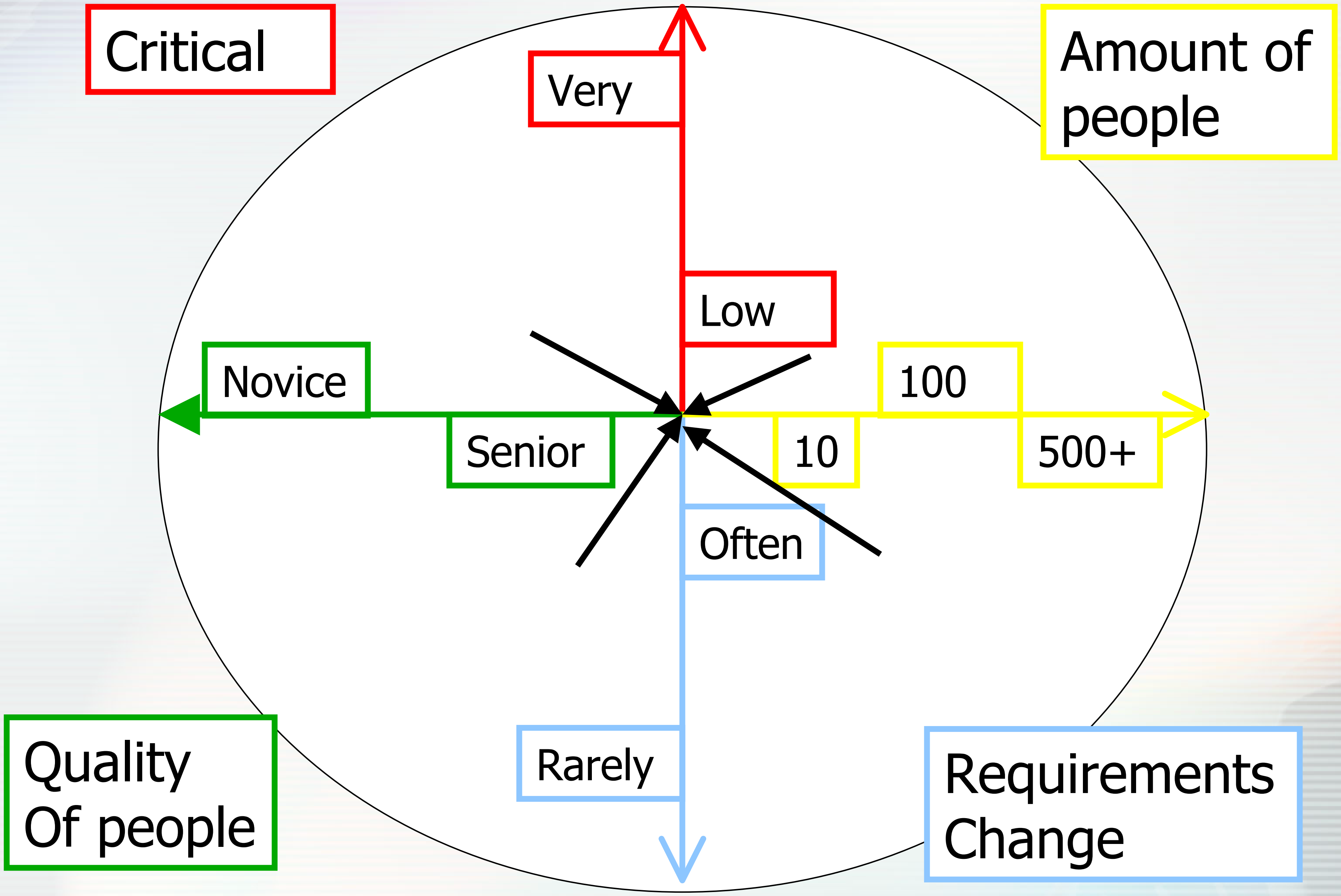
- Agile is adding that arrow.
- Short Iterations – Process
- (Contentious integration – Practice)
- Agile is a family of Software development process methodologies. Big words ☺
- Agile Manifesto.

So, Who’s in the family?

- Feature Driven development
- eXtreme Programming (XP)

  (which is the most common)
- Scrum
- DSDM
- Crystal Clear
- Agile RUP – AUP
- ASD
- …

When do we use XP?



When do we use XP?

- Senior & experienced developers
- Small number of developers
- Low critically
- High Requirements change

XP Critics say…

- **requires too much cultural change to adopt**
- **insufficient structure and necessary documentation**
- **only works with senior-level developers**
- **can lead to more difficult contractual negotiations**

- wikipedia, “Agile software development”

XP Is not enough for some

- Team size < 10
- Very experienced developers
- Low critically
- Very big changes
- Process Buy-in is a must
- There is no golden hammer

Common Agile problems & Solutions

- Non-experienced developers
  - → More process
- High critically
  - → More upfront design
- Big teams
  - → More role definitions
- Change is not an option
  - → Less Change, More adapting
- MORE

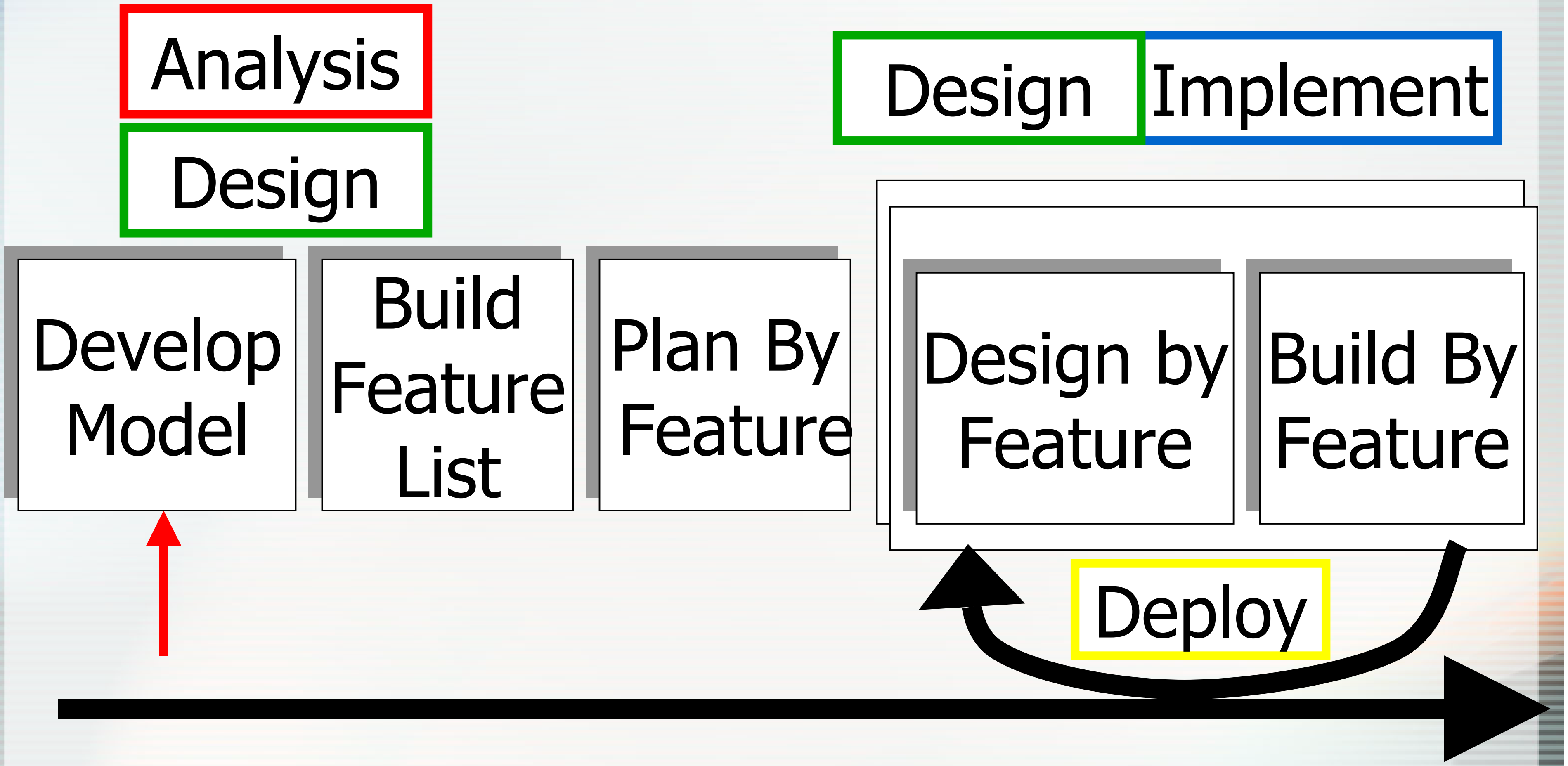
Feature Driven Development

- Feature Driven Development (FDD) can be implemented with:
  - up to 500 developers
  - More critical projects
  - Bigger projects
  - More novice developers
  - Environments that demand Waterfall
- Every methodology has:
  - Process
  - Best Practices

The Three Faces of FDD

- Waterfall
- Extremely Agile
- myFDD
- The boss doesn’t have to know we’re Agile.
- The developers don’t need to know they’re Agile.
- No change = Good.

The FDD Process



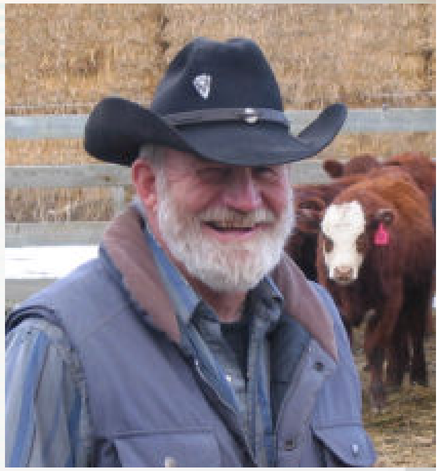
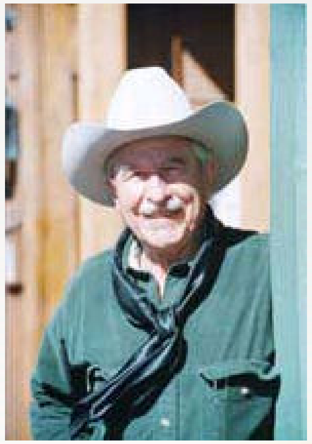
Develop Model

- **Roles we need to assign:**
  - **Chief Architect**
  - **Chief Developers**
  - **Domain Experts (Billy-bob-joe)**

1. **<u>Create Modeling Team</u>: Roles mentioned above & rotating developers.**
2. **<u>Domain Walkthrough:</u> Domain Experts tell us everything they know.**
3. **<u>Study Documents</u>**

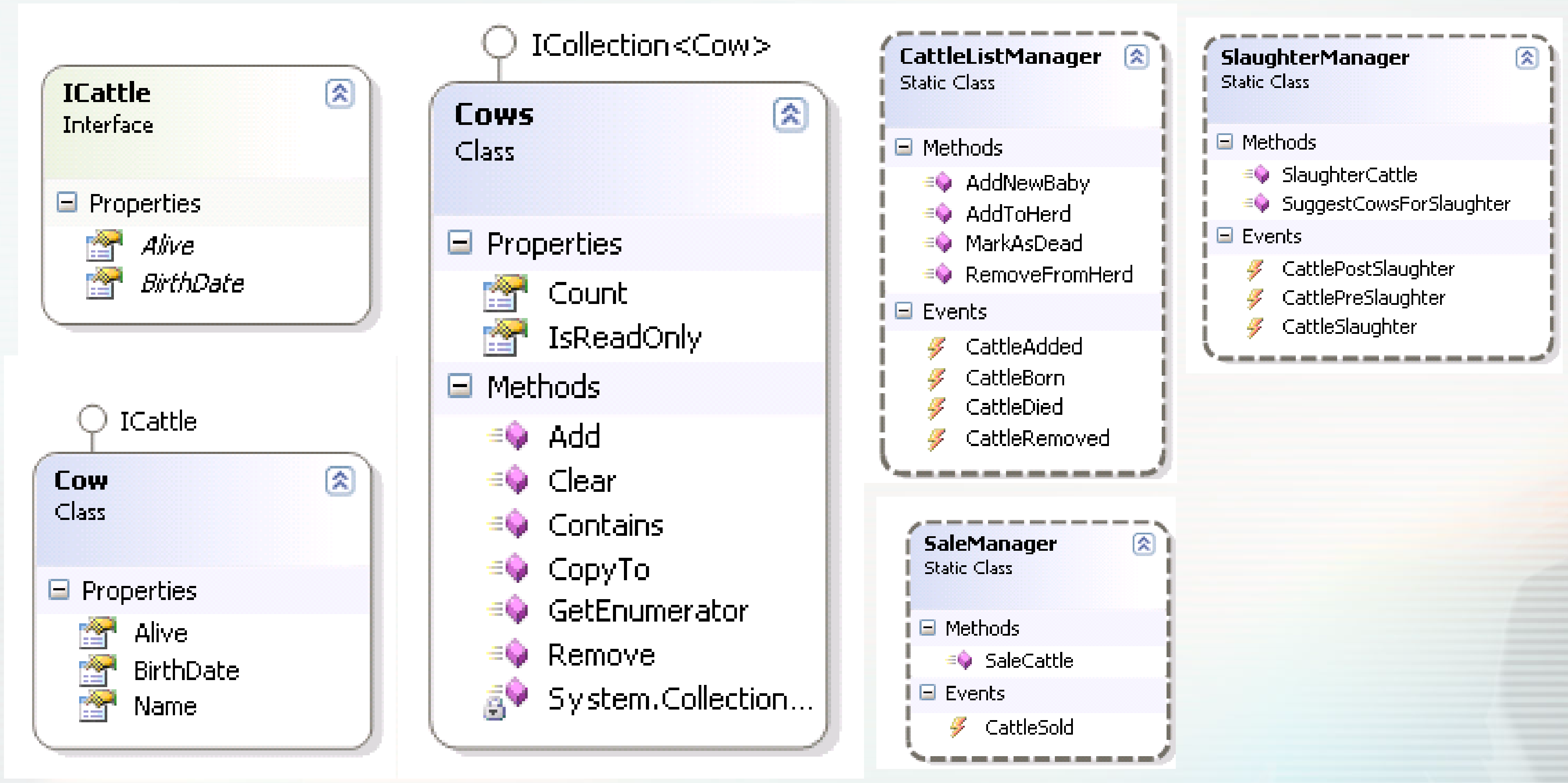
COWS!!!

- Billy-bob-joe is a southern cattle-rancher and he needs a system to manage his farm.
- The system should manage:
  - Existing cattle
  - Breeding
  - Slaughtering & selling meat
  - Selling cattle
- This is our problem domain.

Develop Model - example

4. <u>Create Model</u> in groups of three people. How about this one?



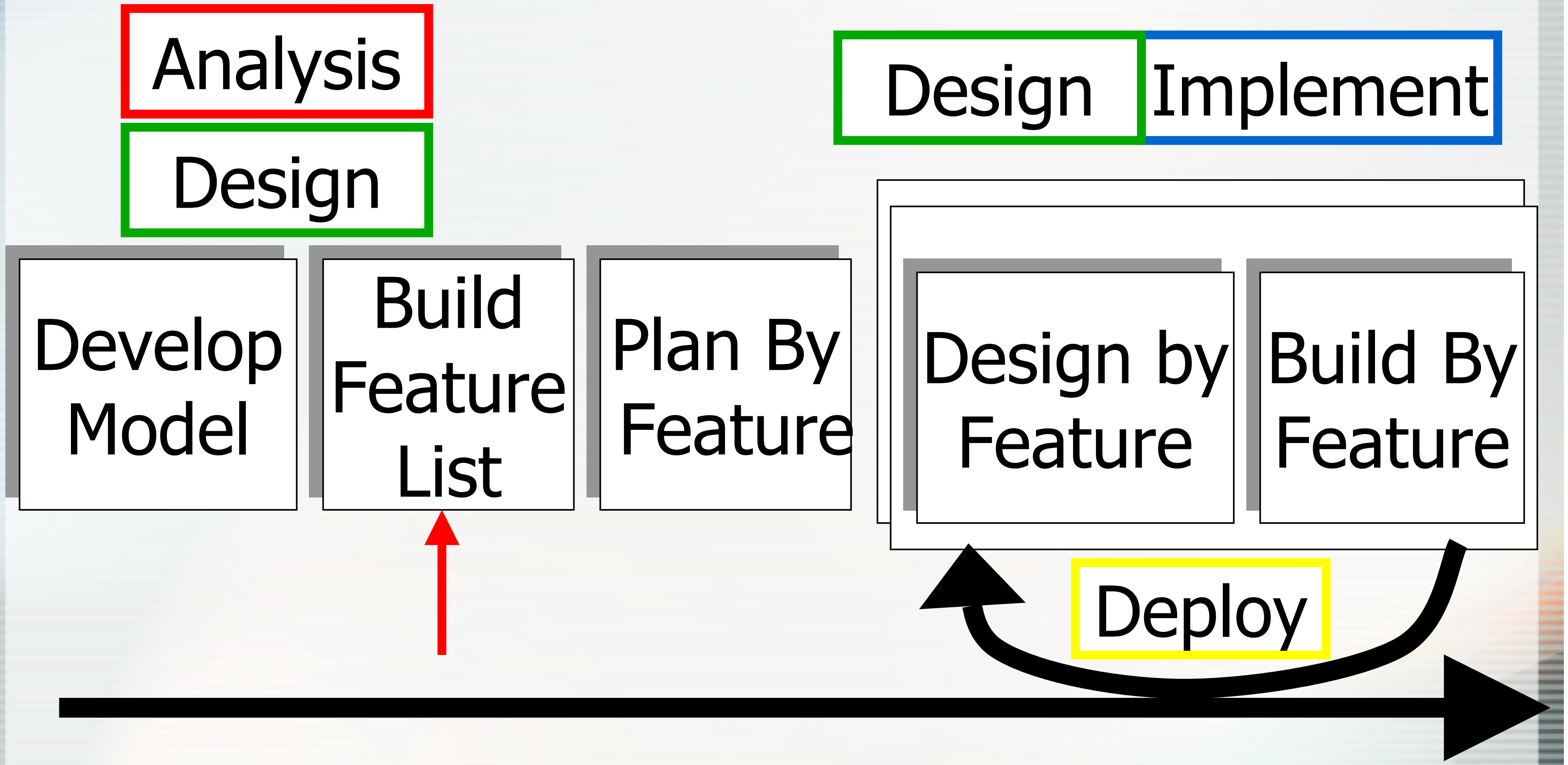
Develop Model – Straw man

- **What did you think of that Model?**
- **This was intentionally a “weak” Model the chief architect created.**
- **“Straw-man” Model**
- **The groups of three people “captured” my Model and while doing so improved it and exposed it’s weak points.**

Develop Model – Three Faces

- **Alternative Models as notes.**
- **Model Driven Architecture.**
- **Develop Model as Waterfall – 98%-100% complete Model.**
- **Develop Model as extremely Agile – 60%.**
- **myFDD should be about 70%-80%.**

The FDD Process



Build Feature List

- Do one thing – Build a Feature List.
- A FDD “Feature” is a small client valued feature.
- Small
- Client
- Valued
- Feature
- Feature - <action> <result> <object>
- Feature Set – <action>ing <object>
- Major Feature Set – <object> Management

Build Feature List - Example

- Feature - <action> <result> <object>
- Feature Set – <action>ing <object>
- Major Feature Set – <object> Management

Herd Management

Birthing cattle

1. Add a new baby Cattle To Herd
2. Mark Mom not pregnant for Cattle

Slaughter Management

Slaughtering Cattle

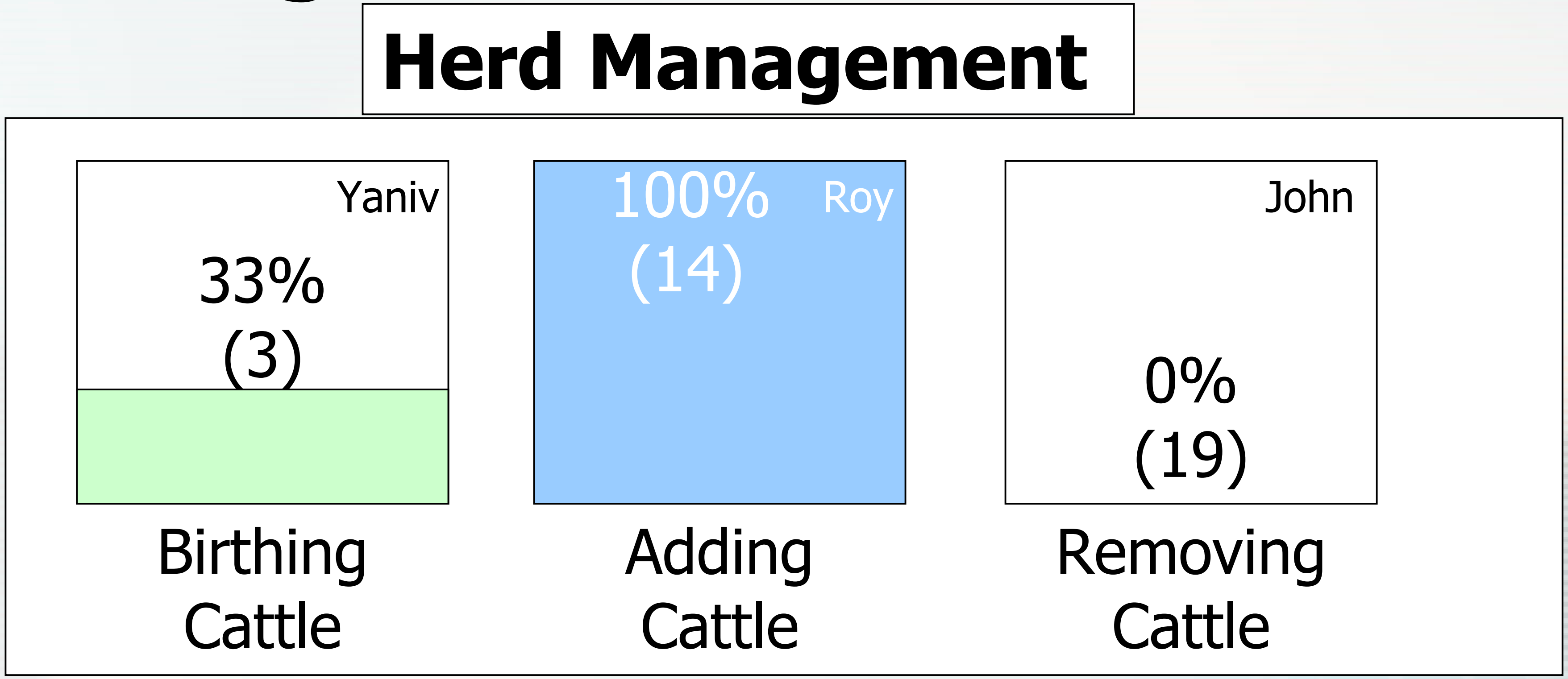
3. Calculate Price For Cattle
4. Add meat to Meat Storage
5. Remove Dead cow from herd.

Build Feature List – Feature & Model

- Feature ←→ Model
- Add a new baby Cattle To Herd ←→ Herd.AddNewBabyCattle(Cattle)
- Mark Mom not pregnant for Cattle ←→ Cattle.MarkMomAsNotPregnant
- Calculate Price For Cattle ←→ Cattle.CalculatePrice
- Add meat to Meat Storage ←→ MeatStorage.AddMeat(Meat)

Build Feature List - Reports

- Features are reportable!
- Client is always informed.
- Management also has access ☺



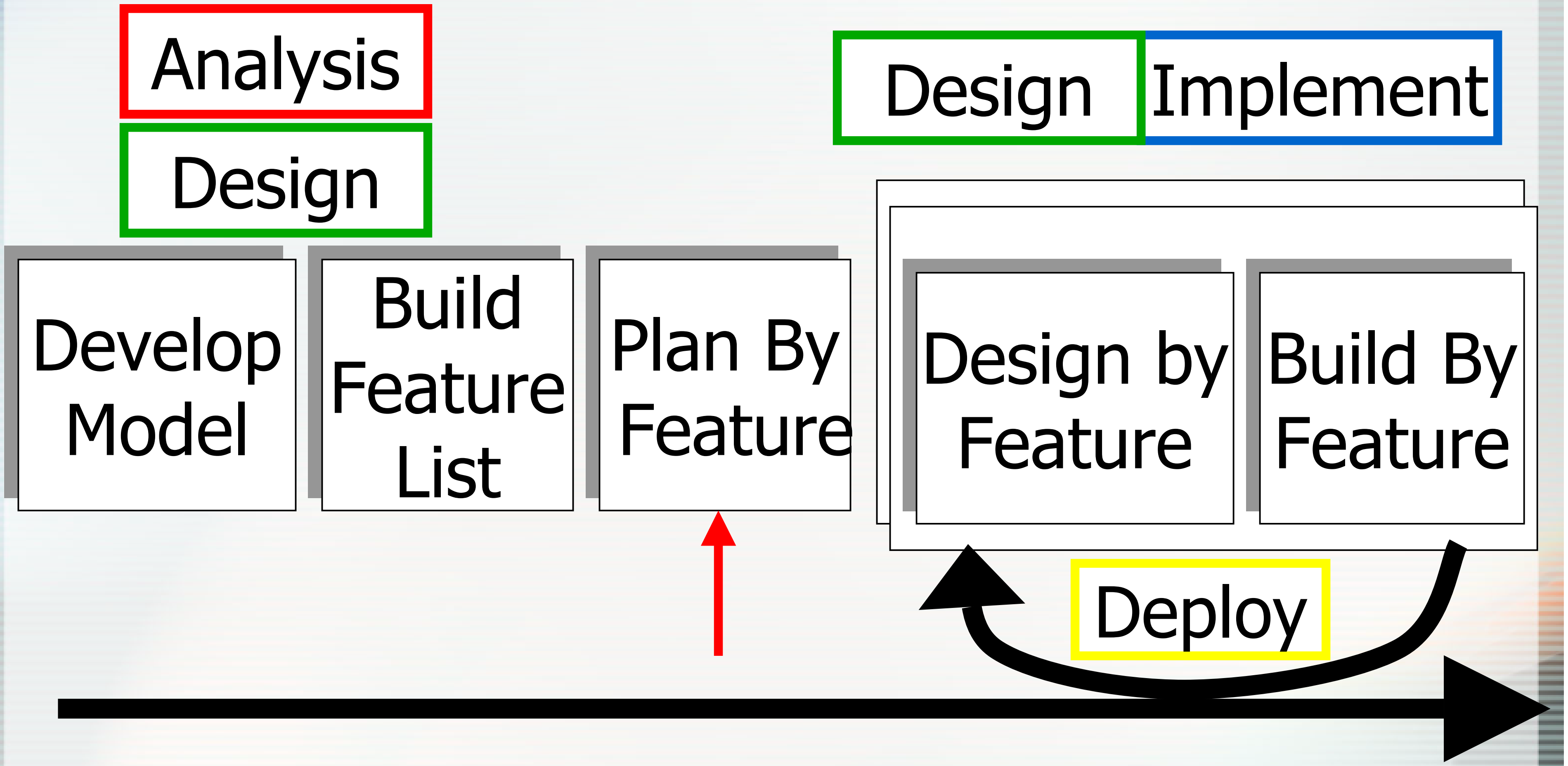
Build Feature List – Summery

- Features are small & client valued.
- Feature list is very short.
- Reportable.
- Testable.
- Feature sets are Assignable.
- Feature sets → iterations.
- Iterations can be planned.

Build Feature List – Three faces

- Features as waterfall – write up 95%-100% of the features and sign as contract.
- Features as Extremely agile – 70-80%.
- myFDD – 80%-90%

The FDD Process



Feature Sets into iterations

1. **Determine Development Sequence**
   - **Check Dependencies (cow before cows)**
   - **Consider High-risk feature**
   - **Consider High complexity features**
   - **Either by Date or by Sequence.**
2. **Assign Project Manager**
3. **Assign Chief developers to feature sets**
4. **Assign developers as Class Owners**

Plan By Feature - Example

Justin → "Meat Storage" Class Owner

Miki → "Cow" Class Owner

Oren → "Herd" Class Owner

Roy → "Slaughtering Cattle" Chief Developer

Yaniv → "Birthing Cattle" Chief Developer

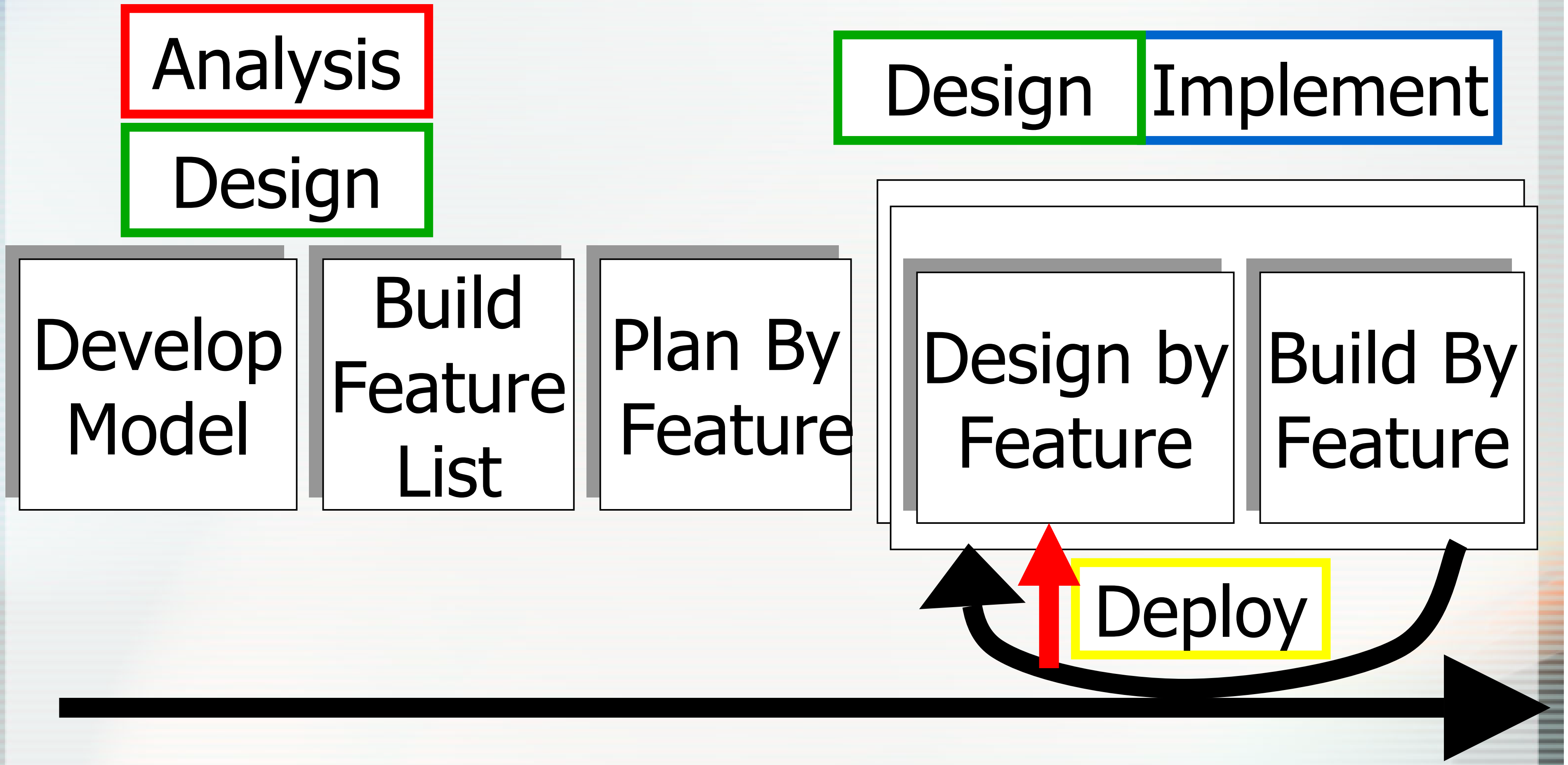
Feature Sets into Iterations:

1. Adding Cattle, Removing Cattle
2. Birthing Cattle, Killing Cattle
3. Storing Meat, Selling Cattle
4. Slaughtering Cattle, Selling Meat

Plan by feature

- Planning like waterfall – Set dates for the completion & start date, hours to work and for each feature set.
- Planning extremely Agile – determine the order of Feature sets.
- Planning myFDD – determine completion months for feature sets.
- Anyway – group Feature Sets into Iterations.

The FDD Process



Design By Feature

- This is the first part of short iteration.
- We know which feature sets we need to build.
- Now it’s time to design the software we will build.