Welcome

Evolutionary Methodology – What?

Group Survey

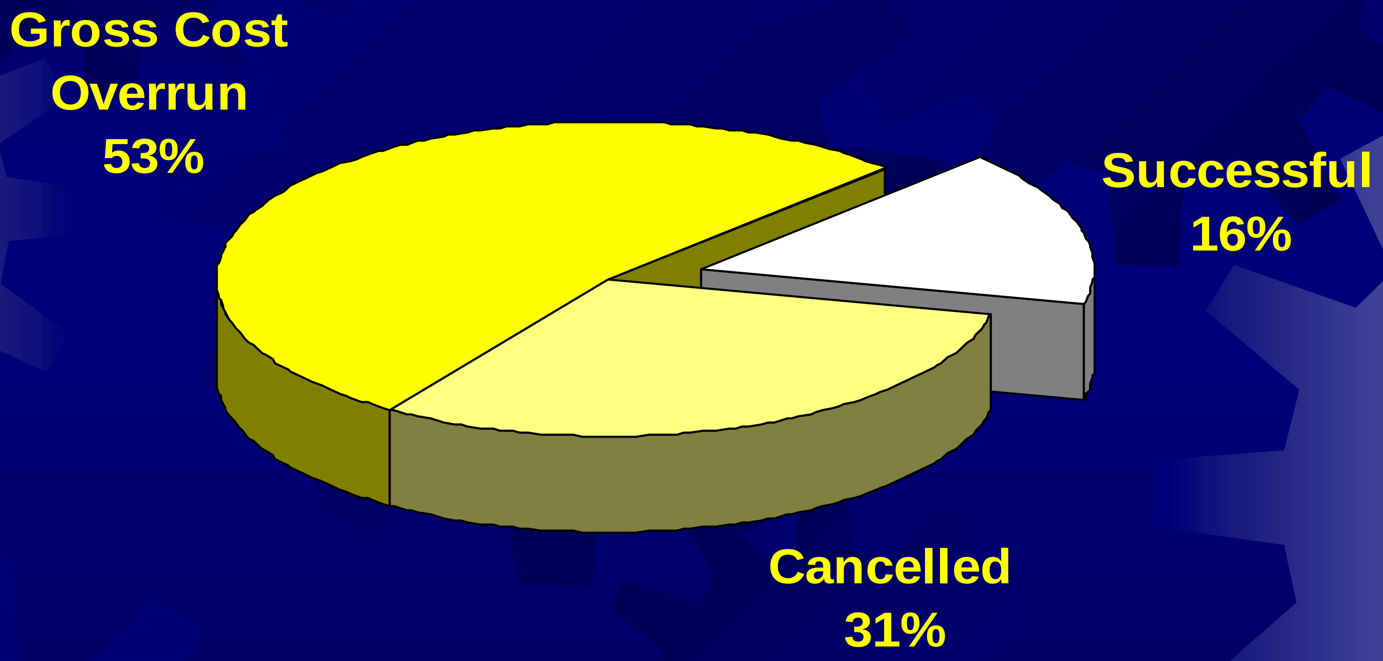
- How many have been involved with IT development projects in last 12 months
  - How many were successful
- Number in the IT organization? Business?

Industry Trends

- Large single projects fading away (US)
- Cultural shift to teams of < 10 people and project duration of < 2 years
- Increased user involvement
- Increased process focus
- Increased use of component-based software
- Self-managed project teams

Source: Ed Yourdon, C/S Developers Conference

Software Development *Reality...*



Source: Application Development Trends

Business Reality...

- Rapidly changing technology
- Shortened time to market
- Changing or unknown requirements
- Systems meet requirements but don’t do the job
- Unwillingness to invest in large projects
- Frozen or shrinking IS budgets and staffing
- Loss of faith that IT still provides value

Evolutionary Development Addresses Today's Business Reality...

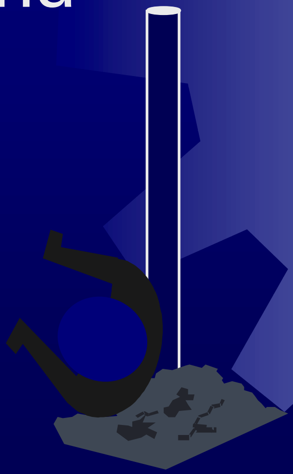
- An iterative development process,
- Scaled for small teams,
- That delivers useful functionality
- In about 90 day increments.

Success requires...

- Incremental development
- Strong client sponsor
- User-centered analysis and design
- Experienced team / management
- High performance teams
- High productivity tools
- Prototyping
- Timeboxed development

“Good Enough” Engineering

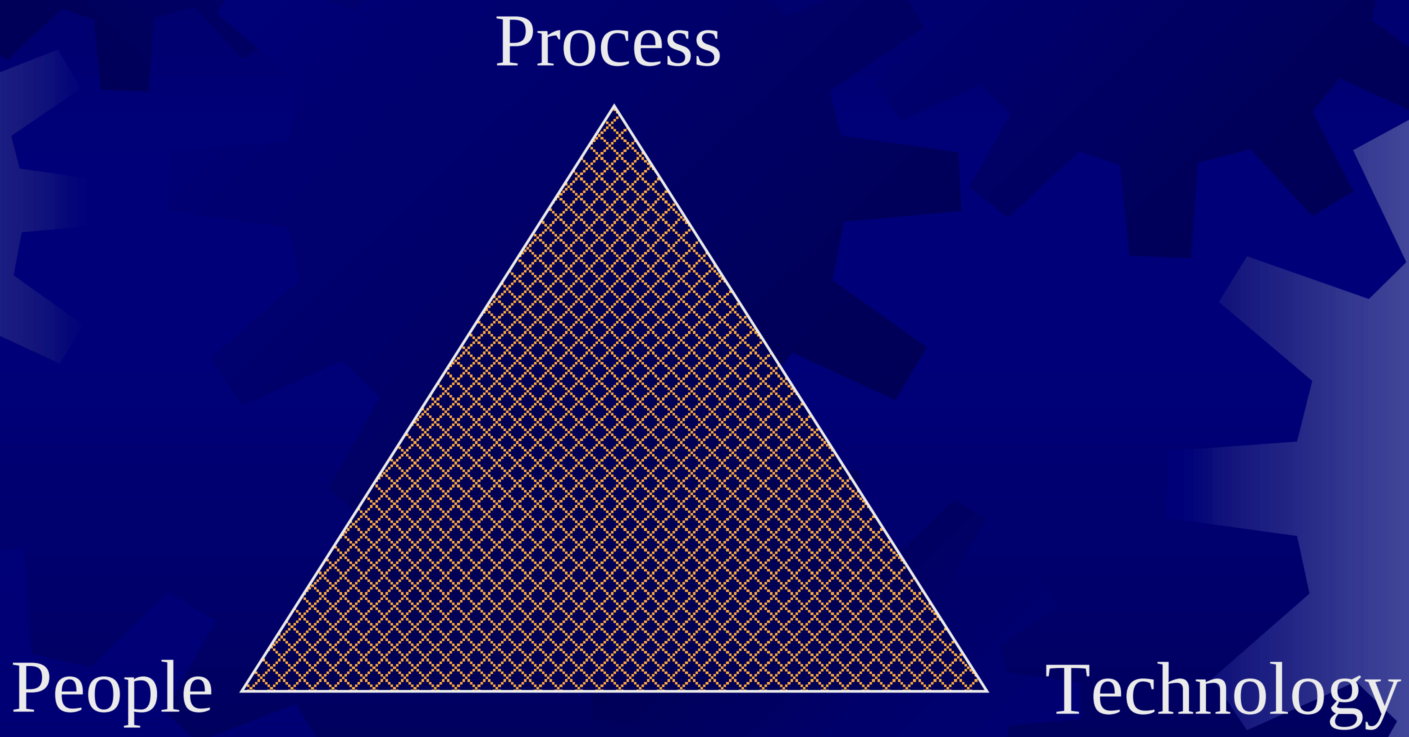
- Today’s businesses can’t tolerate “zero defect software”
- What users really want is software that’s cheap enough, fast enough, functional enough, and built soon enough!
- Requires very different behaviors and strategies for IS shops
- 80/20 rule



Methodology is a project management issue...

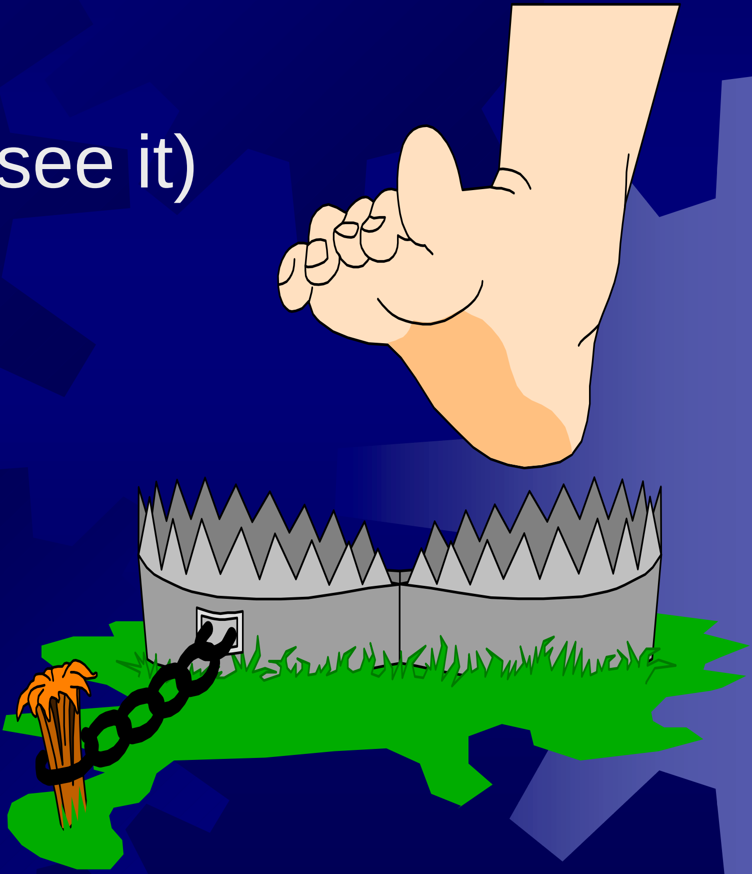
- Not a technology issue
- Success is typically not a technology issue either

Methodology Concepts

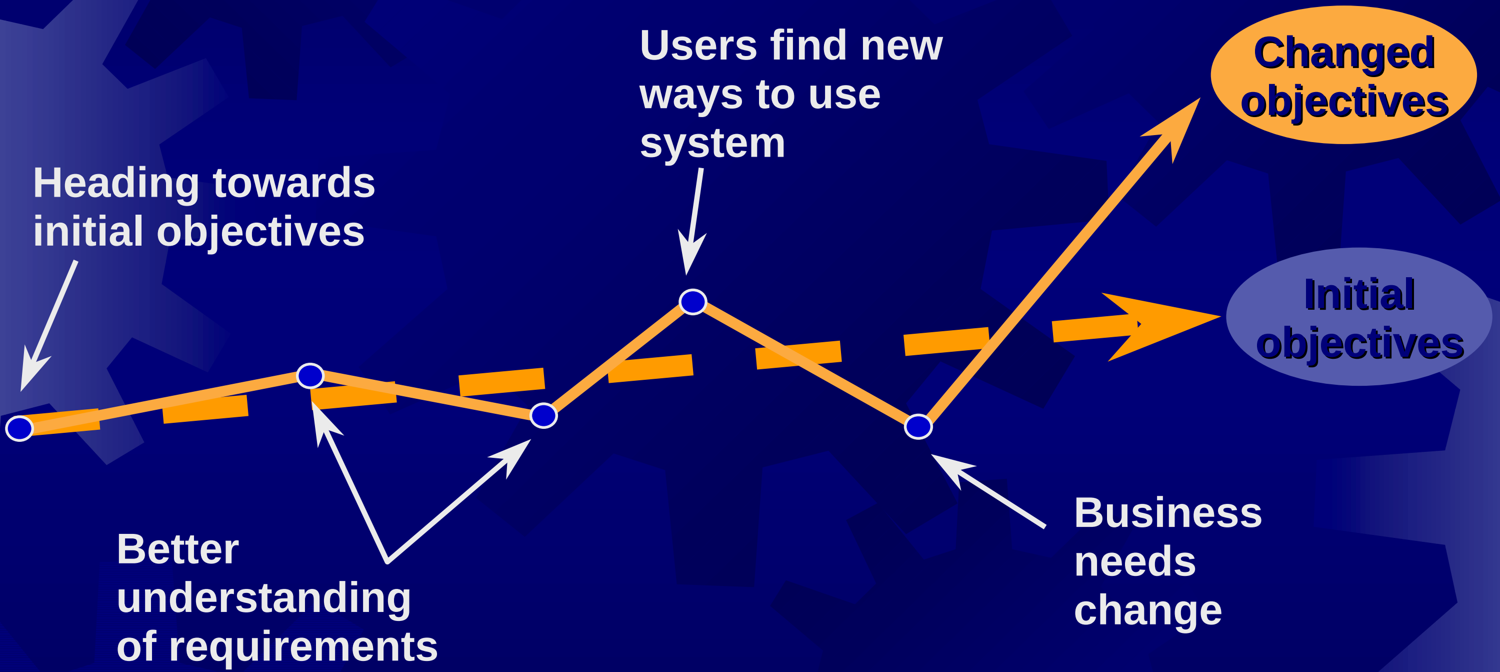


Approaches often confused with Rapid Delivery Methodologies

- Hacking
- IKIWISI (I'll know it when I see it)
- SOTP (Seat of the pants)
- DITIFI (Do it, try it, fix it)
- Nike
- Prototyping

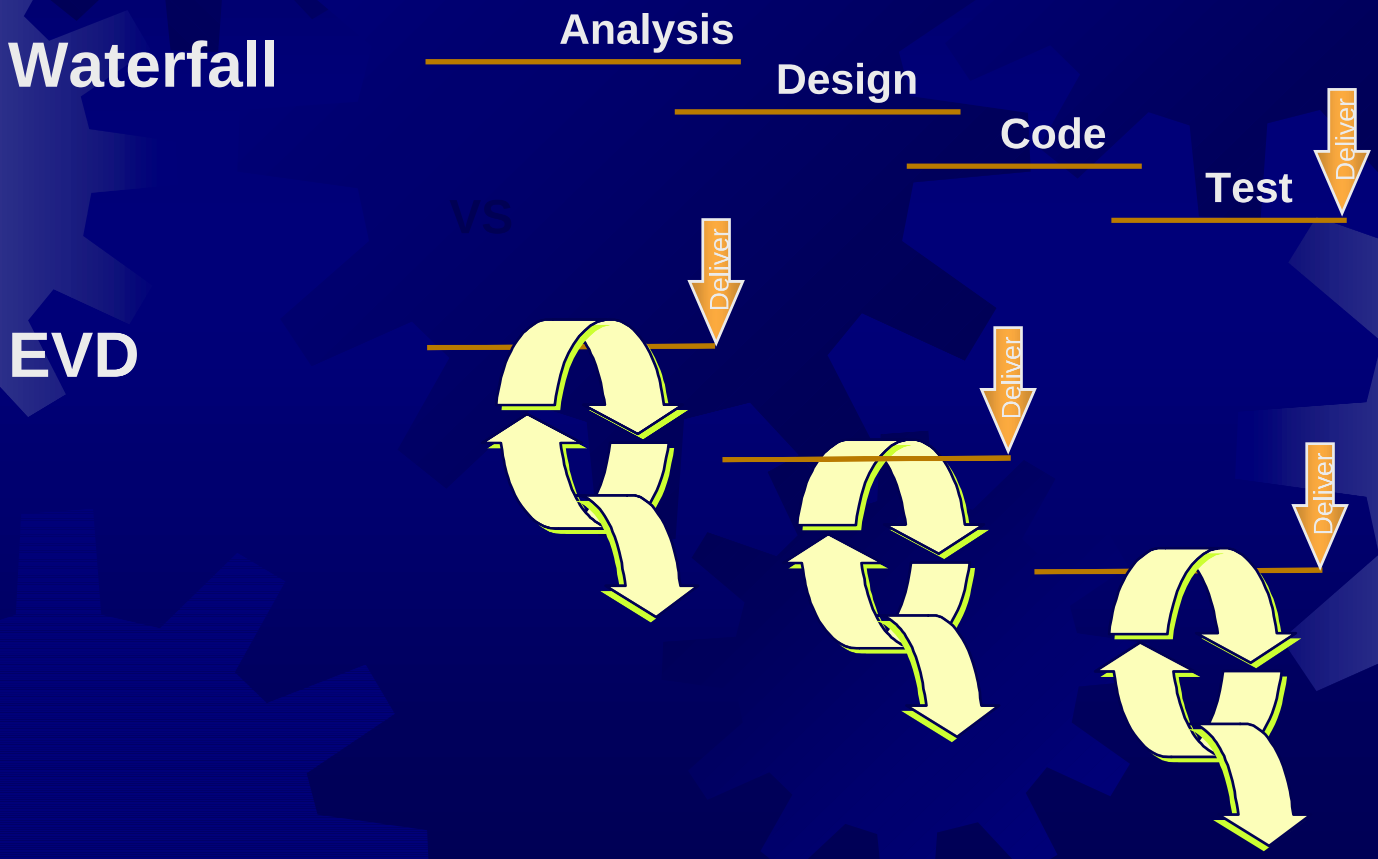


Evolutionary development and delivery


Users find new ways to use system
Changed objectives
Heading towards initial objectives
Initial objectives
Better understanding of requirements
Business needs change

An evolutionary approach

“EVD” compared to the traditional life cycle



Waterfall

Analysis

Design

Code

Test

Deliver

vs

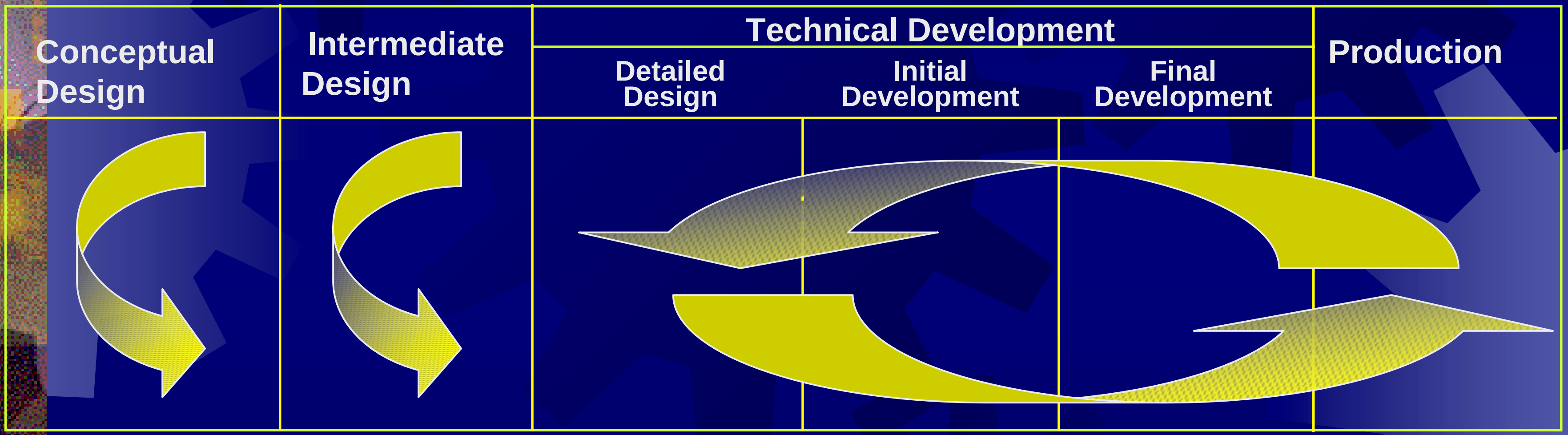
EVD

Deliver

Deliver

Deliver

Overview of the Development Process

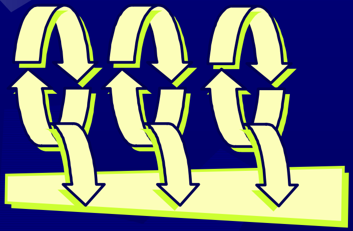


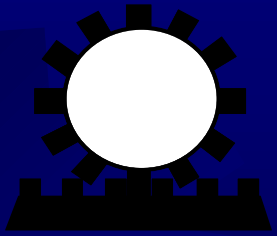
Useful Functionality!

EVD methodology, a closer look

Daman's definition

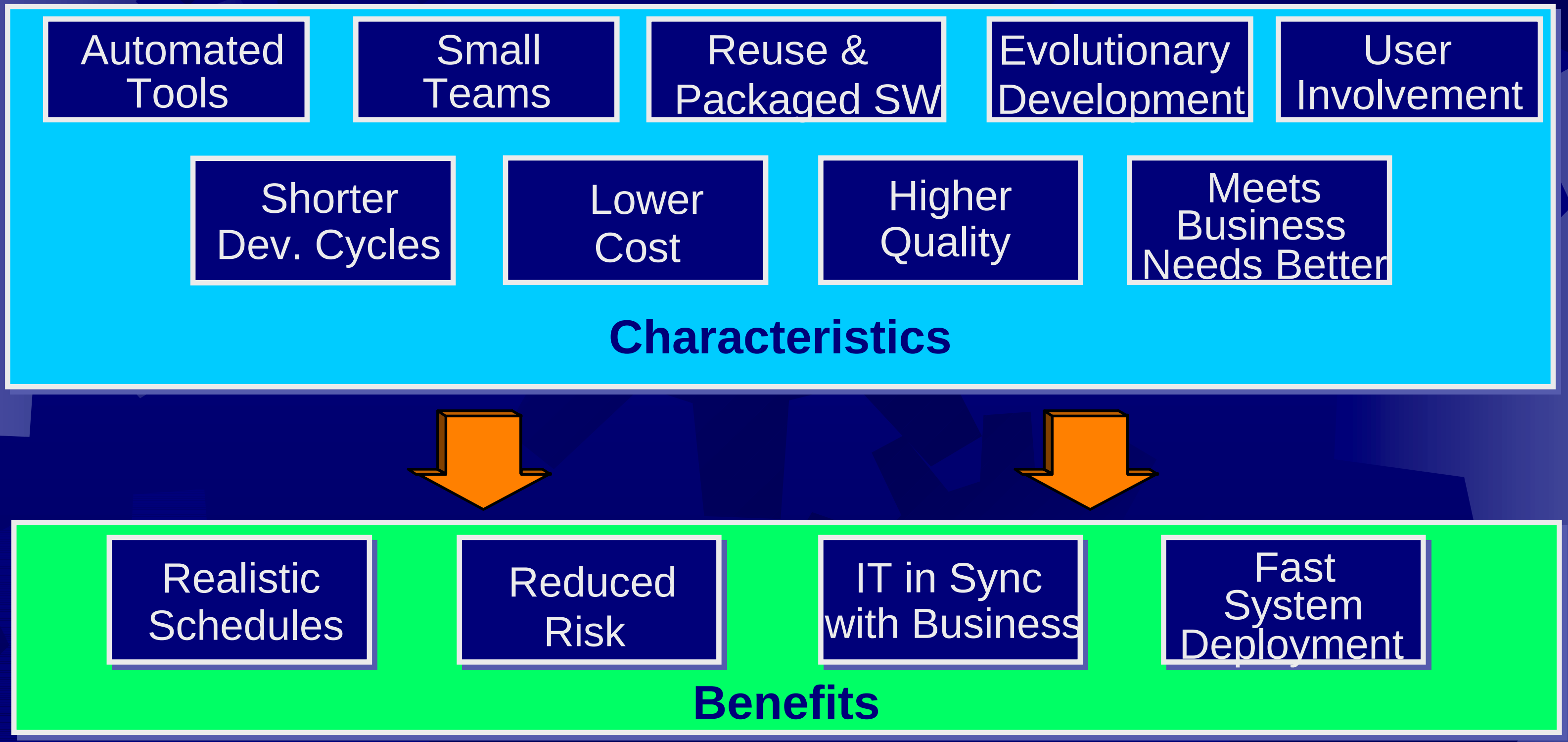
An evolutionary development process, scaled for small teams using appropriate technology, that delivers useful functionality in about 90 days increments.





Process + People + Technology

Benefits of EVD



Automated Tools

Small Teams

Reuse & Packaged SW

Evolutionary Development

User Involvement

Shorter Dev. Cycles

Lower Cost

Higher Quality

Meets Business Needs Better

Characteristics

Realistic Schedules

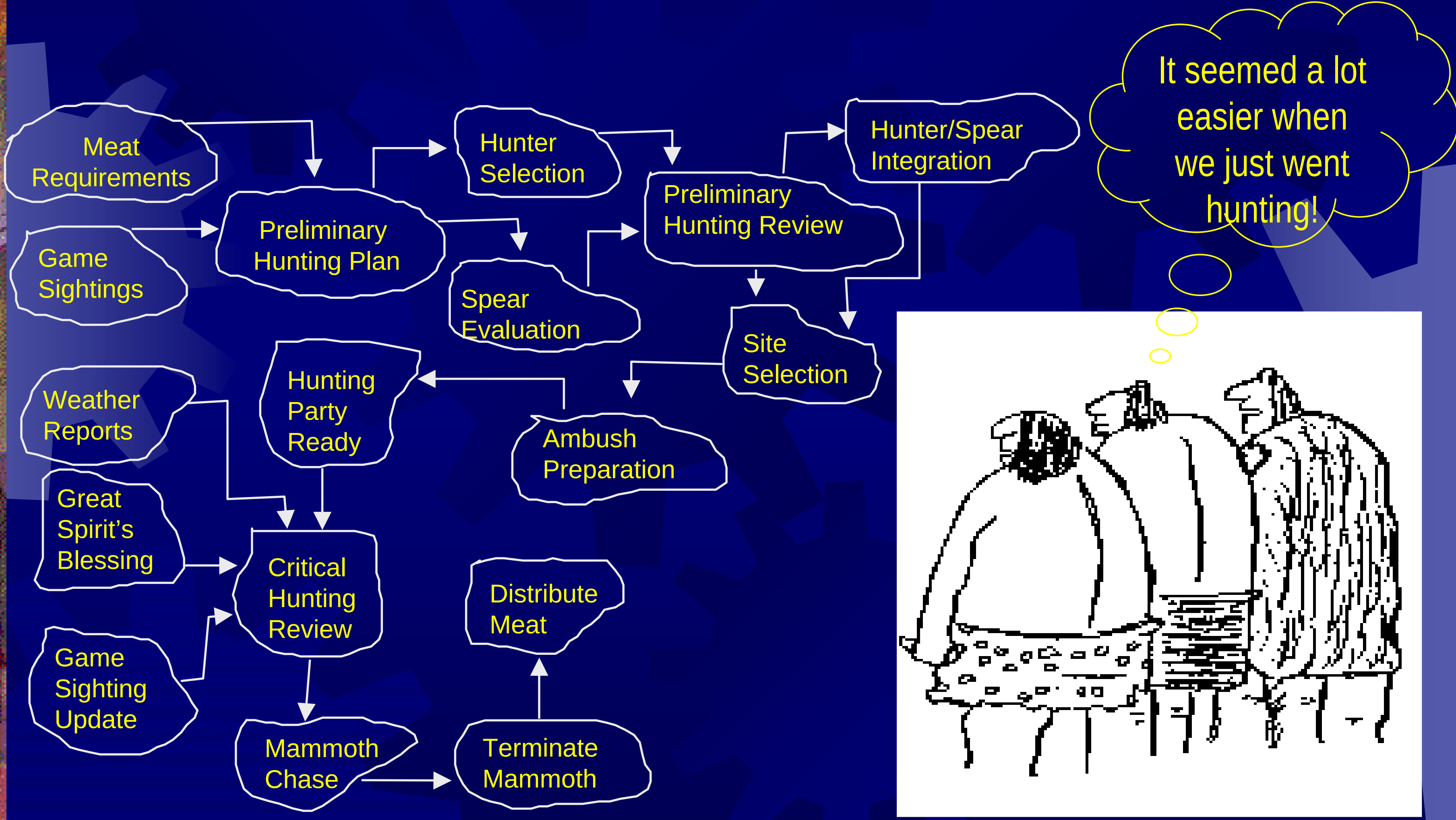
Reduced Risk

IT in Sync with Business

Fast System Deployment

Benefits

Life Cycles:
Why the Neanderthal Became Extinct



Meat Requirements

Game Sightings

Preliminary Hunting Plan

Hunter Selection

Spear Evaluation

Preliminary Hunting Review

Hunter/Spear Integration

Site Selection

Weather Reports

Hunting Party Ready

Ambush Preparation

Great Spirit's Blessing

Critical Hunting Review

Game Sighting Update

Distribute Meat

Mammoth Chase

Terminate Mammoth

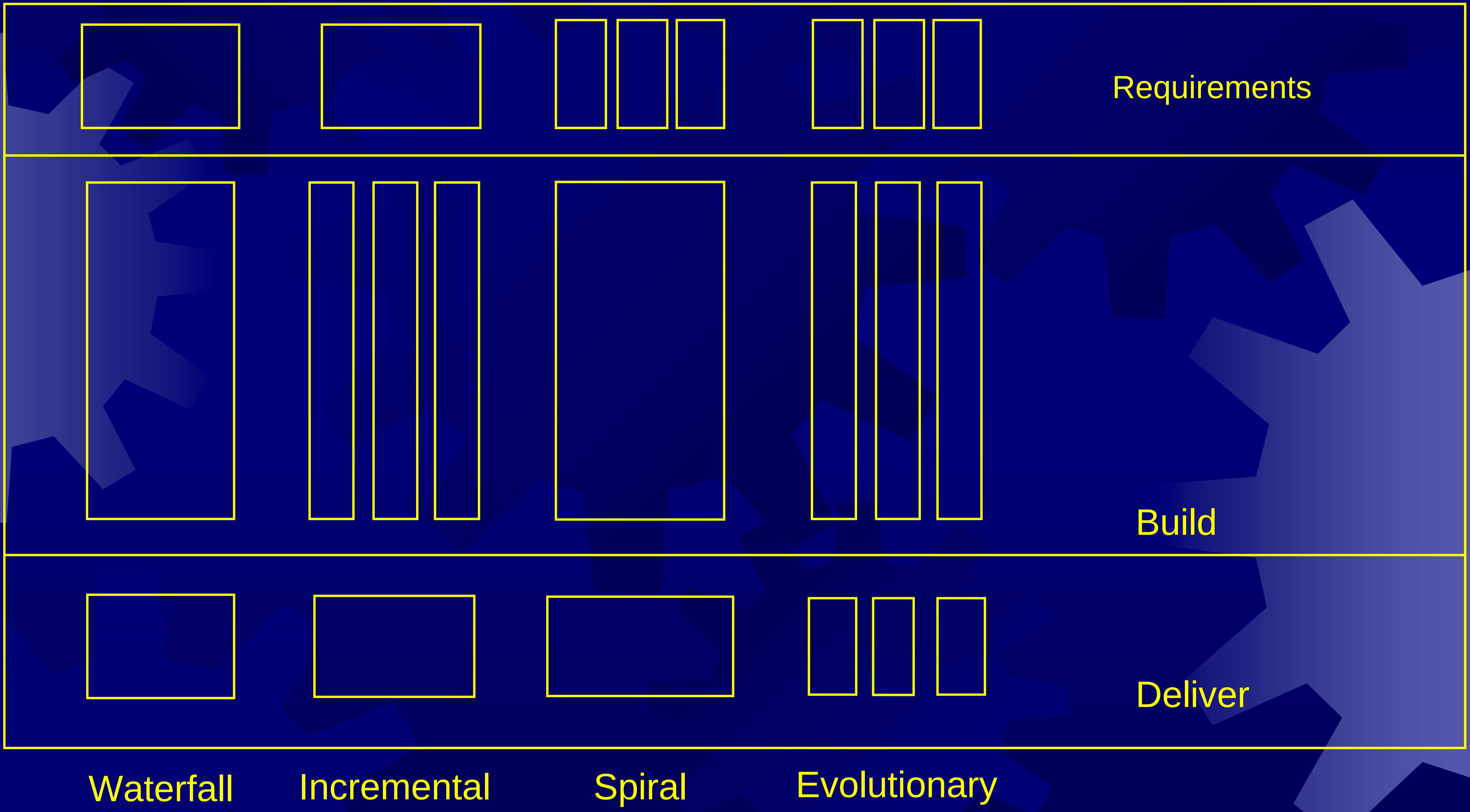
What is a Life Cycle?

- Combination of Methods, Tools, Procedures
- Methods
  - Establish principles of development
  - Describe typical activities
  - Imply a sequencing & frequency of activities
  - Provide guidelines and rules of thumb
- Tools
  - Simply automate methods
- Procedures
  - The management means to sequence project

Why Have a Life Cycle?

- Life cycle map
- Answer questions
- Enable repeatability and consistent deliverables
- Lower risk, increase chances for success
- Easier staff transitioning
- Way to deal with complexity

Life Cycles: Families



Multiple Life Cycles

Using the *appropriate* life cycle can achieve a delivery rate 4 - 5 times faster than a traditional approach.

Comparing Life Cycles

	Waterfall	Incremental	Spiral	Evolutionary	Rapid EVD
Known Requirements	Yes	Yes	No	No	No
Single Build	Yes	No	No	No	No
Single Delivery	Yes	No/Yes	Yes	No	No
Prototyping	No	No	Yes	No	Yes

Comparing Life Cycles

	Waterfall	Incremental	Spiral	Evolutionary	Rapid EVD
Known Requirements	Yes	Yes	No	No	No
Single Build	Yes	No	No	No	No
Single Delivery	Yes	No/Yes	Yes	No	No
Prototyping	No	No	Yes	No	Yes

Selecting A Life Cycle

Factor	Waterfall	Incremental	Spiral	Evolutionary	Rapid EVD
Project Experience	Low	Low	High	High	High
Volatile Requirements	No	No	Yes	Yes	Yes
Project Risks	Low	Medium	High	High	High
Project Size	Small-Medium	Medium	Small-Large	Small-Large	Small
User Involvement	Low	Low	Medium	High	High

Why Methodologies are Abandoned

- Lack of management support/commitment
- Don’t match the way people need to work
- Don’t perceive any value
- Lack of education
- Treated dogmatically, too rigid
- Customers don’t want to pay for “overhead”
- Myths

An example of EVD Project Tasks

Conceptual Design	Intermediate Design	Technical Development			
		Detailed Design	Initial Development	Final Development	Production
• Define Team • Target • Success Measures • Verify Customer needs & Segments • Model biz process • Model data entities • Prototype UI (EVD or throw-away) • JAD	• Develop Arch Framework • Map biz process to objects • BOLD Gap analysis • Verify usability • Logical Data Model	• 3rd party s/w •Security •Operations •Physical data model •Object Specs •Full Arch Specs •QA test cases	• Refine & prototype UI (working code) •Refine workflow •Backend integration •Detail QA	• Deliver final product •Biz signoff •Production Support Processes •Train users	• Dev warranty support •Doc Procedures for SMC •Manage system •Update docs

(1 – 4 wks)

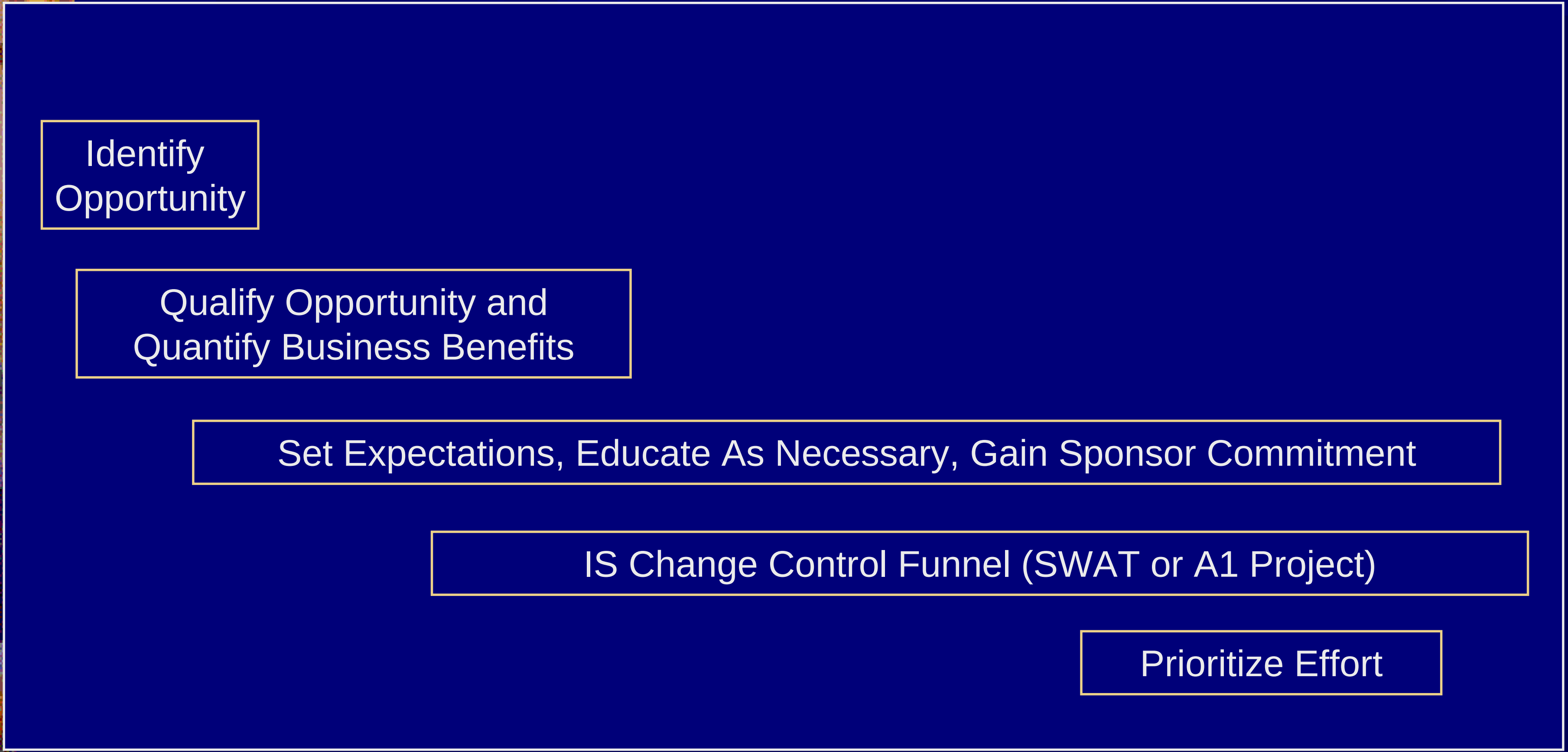
(90 days)

Example EVD Project Documents

Conceptual Design	Intermediate Design	Technical Development			
		Detailed Design	Initial Development	Final Development	Production
• Project Definition • Customer Needs Definition • Biz Process Model • Conceptual Data Model • UI mockups	• Arch Framework • Object Gap Analysis • Logical Data Model • High-level project plan	• Operational Framework • Physical Data Model • Detail Object requirements • Test Plan / test cases • Detailed project plan	• Updated Biz process model • UI first cut	• Updated Project Schedule • System docs • Production handoff doc • System build	• Update Project Definition • Update Biz & Data Models • Lessons Learned • Reuse Catalog • Next Phase Planning

← (1 – 4 wks) → ← (90 days) →

Qualification: Activities



Identify Opportunity

Qualify Opportunity and Quantify Business Benefits

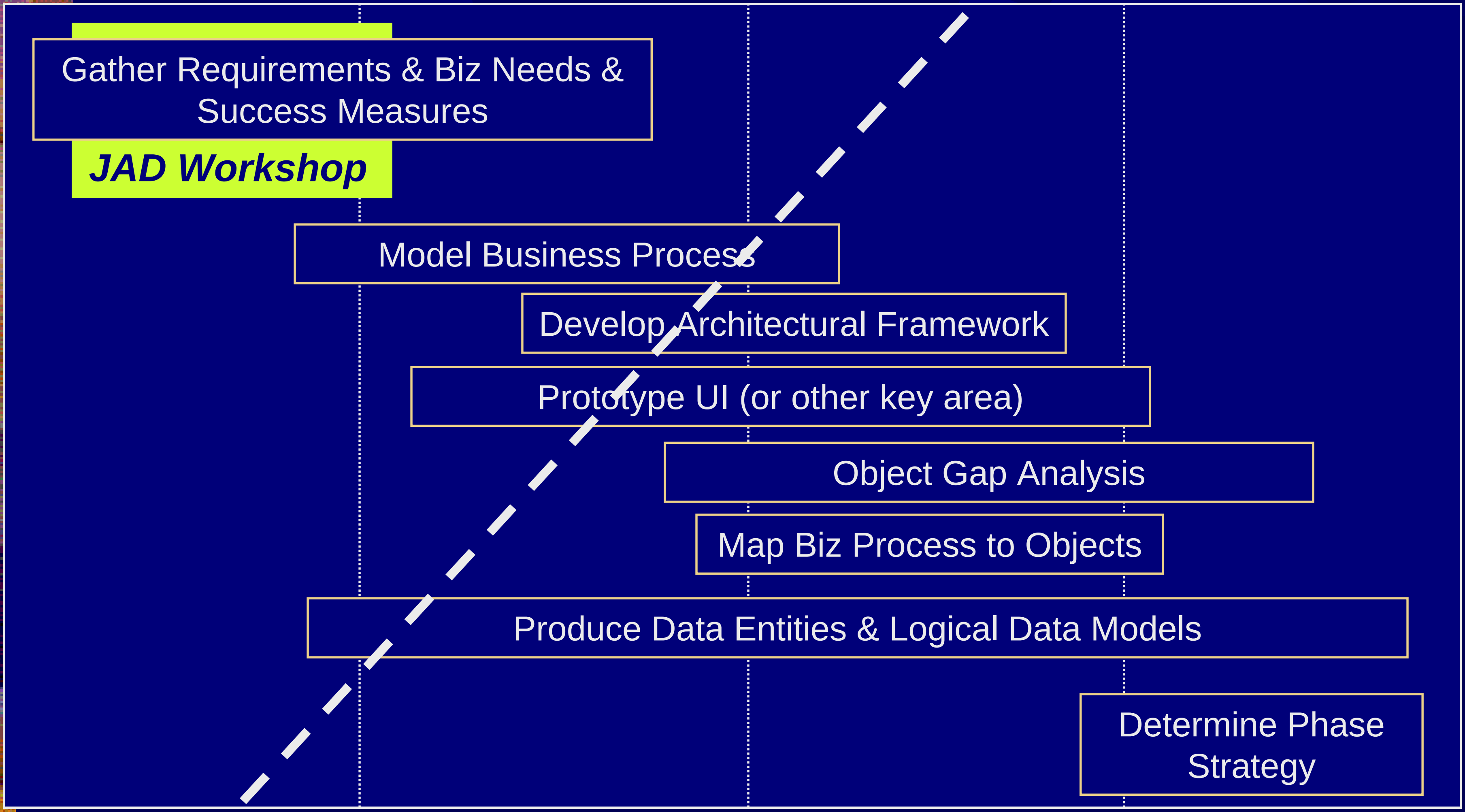
Set Expectations, Educate As Necessary, Gain Sponsor Commitment

IS Change Control Funnel (SWAT or A1 Project)

Prioritize Effort

(1- 14 days)

Definition & Early Design Activities



Gather Requirements & Biz Needs & Success Measures

JAD Workshop

Model Business Process

Develop Architectural Framework

Prototype UI (or other key area)

Object Gap Analysis

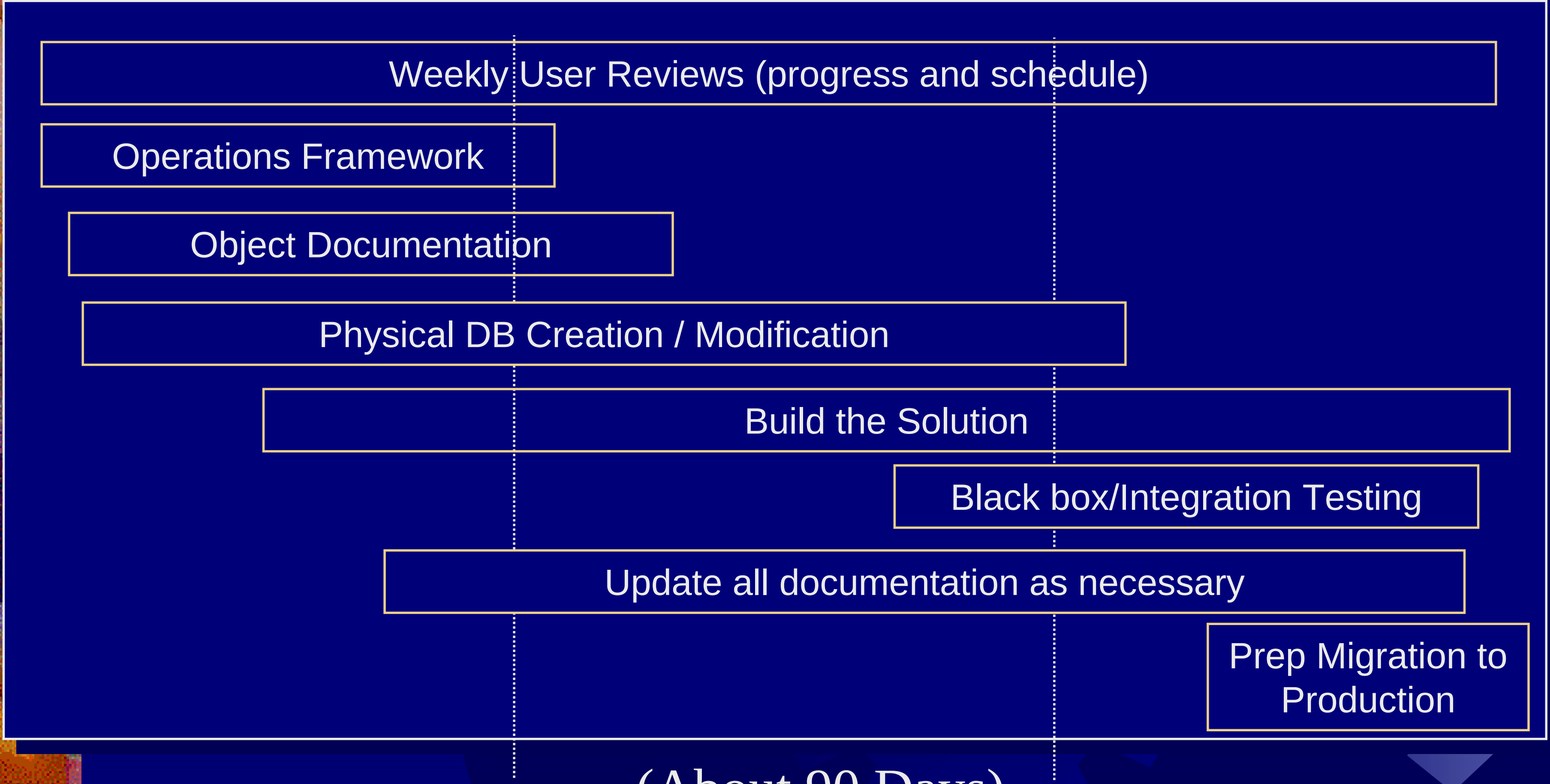
Map Biz Process to Objects

Produce Data Entities & Logical Data Models

Determine Phase Strategy

(1- 4 weeks)

Technical Development: Activities



Weekly User Reviews (progress and schedule)

Operations Framework

Object Documentation

Physical DB Creation / Modification

Build the Solution

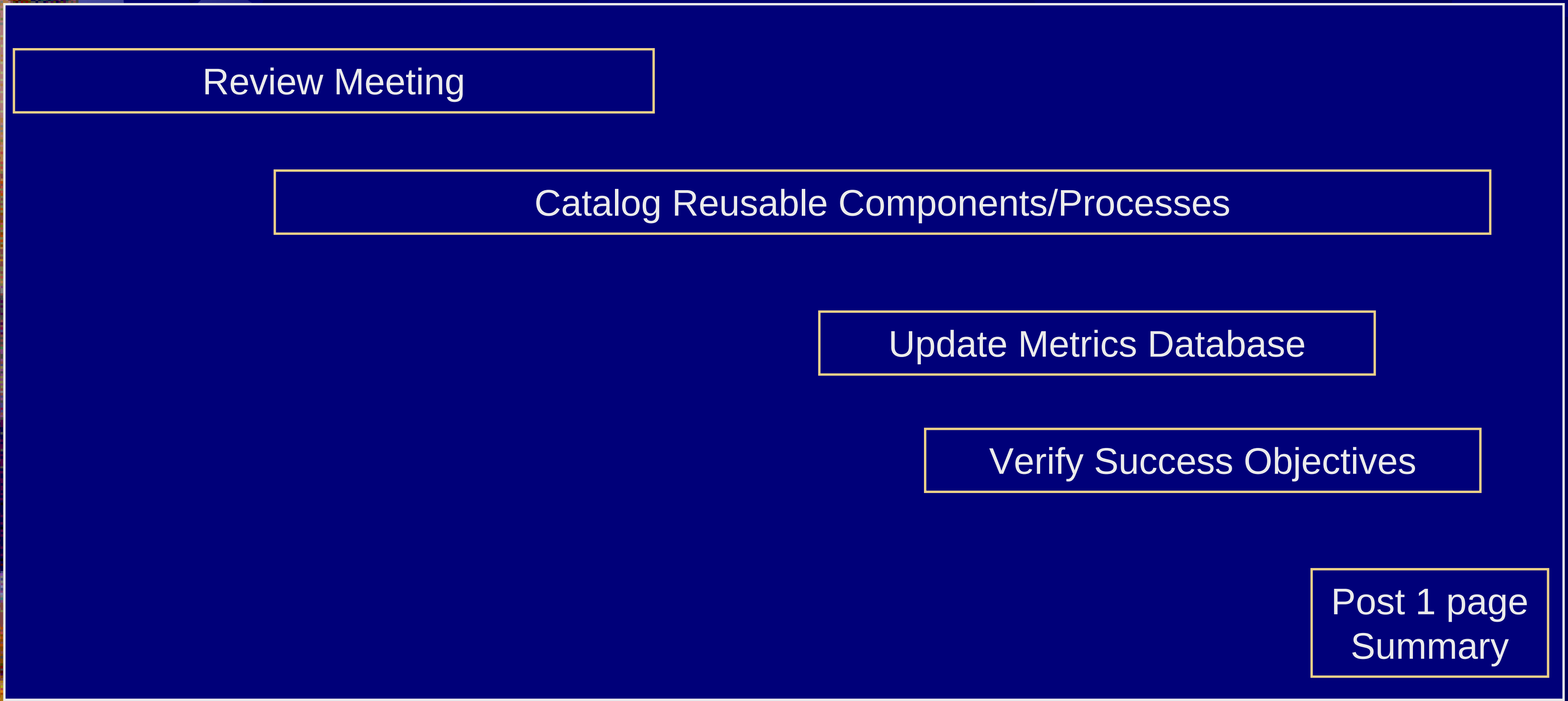
Black box/Integration Testing

Update all documentation as necessary

Prep Migration to Production

(About 90 Days)

Project Wrap Up: Activities



Review Meeting

Catalog Reusable Components/Processes

Update Metrics Database

Verify Success Objectives

Post 1 page Summary

(1/2 - 2 days)

Questions and Discussion

End of formal presentation

Facilitated Workshops
JAD

- Total effort: 3 -5 days
- Structured Agenda (geared toward docs
- Goals:
  - Teambuilding
  - Conflict resolution
  - Information sharing
- Accountability for decisions reached
- Assigned participant roles & responsibilities
- Generates Consensus
- All participants are equal team members

Comparison of JAD VS Traditional DATA Gathering

REQUIREMENT	**JAD Y/N**	**Traditional Y/N**
Define Problem to be solved	**Y**	**Y**
Builds partnership User involvement, commitment	**Y**	**N**
Consensus on priorities for Bus. & Tech. needs	**Y**	**N**
Defines Problem in reduced Timeframe	**Y**	**N**
Information Sharing	**Y**	**N?**
Breaks down large problems into small one	**Y**	**N?**
Define Current State	**Y**	**Y**
Agree on Prototype	**Y**	**N**

INPUT

- Right People
- Expectations
- Facilitator
- Sponsorship

JAD WORKSHOP

OUTPUT

- Have 80% of info/rqmts needed to model solution
- BAMs/Users prioritize requirements
- A △ T (Actual vs. Theoretical)
- Determine what to prototype
- Start filling in Solution Blueprint

Complete in 3-5 days

Rapid Delivery Differences

- Heavy reliance on JAD (for speed and consensus)
- 80% is good enough
- Prioritization of requirements
- Range of implementation
- Identification of dependencies

Analyze Implementation Constraints

- Defines relationships between high-priority requirements
- Identifies range of implementation when full implementation is not possible in initial phase

$$\sum \text{Requirements} = \sum \begin{pmatrix} \text{BusinessFunctions} + \\ \text{ImplementationConstraints} \end{pmatrix}$$