

The **Agile** vs **Waterfall** Myth

Benjamin Hoyt
CEO – 47 Games, Inc.

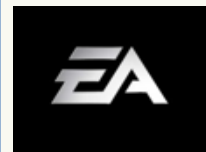
Who is this guy...

- Started Producing Games in 2003
- Interactive Content Producer
- 25+ Shipped Titles & Experiences
- 20th Consecutive GDC*
- Work Experience:

Developers



Publishers



Digital Agencies



...and what has he done?

Console & Mobile (Internal Production)



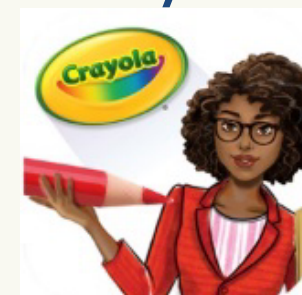
Console & Mobile (External Production)



Location-Based Experiences



AR/VR



Crypto/DeFi



Commonalities

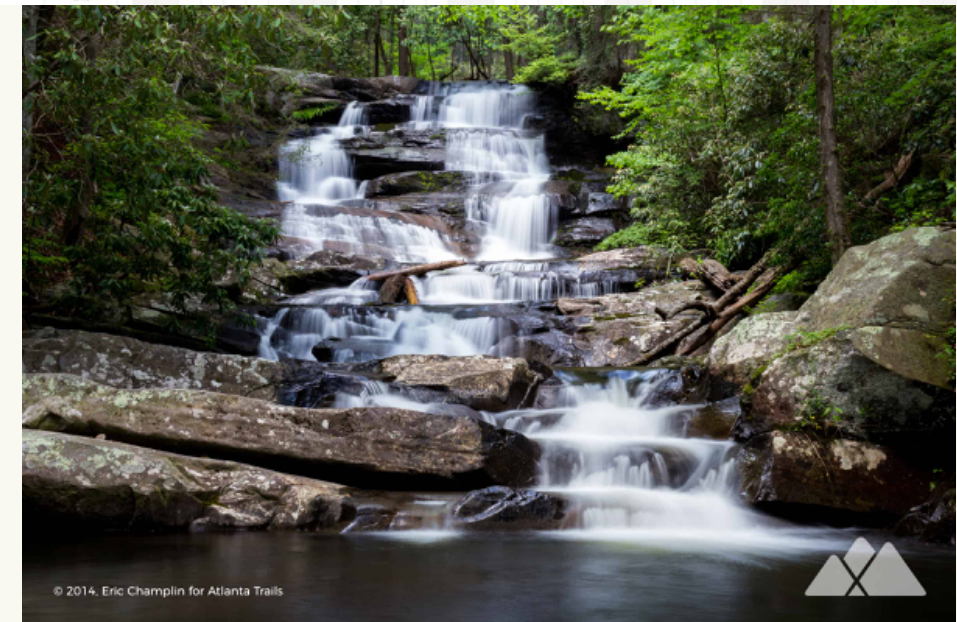
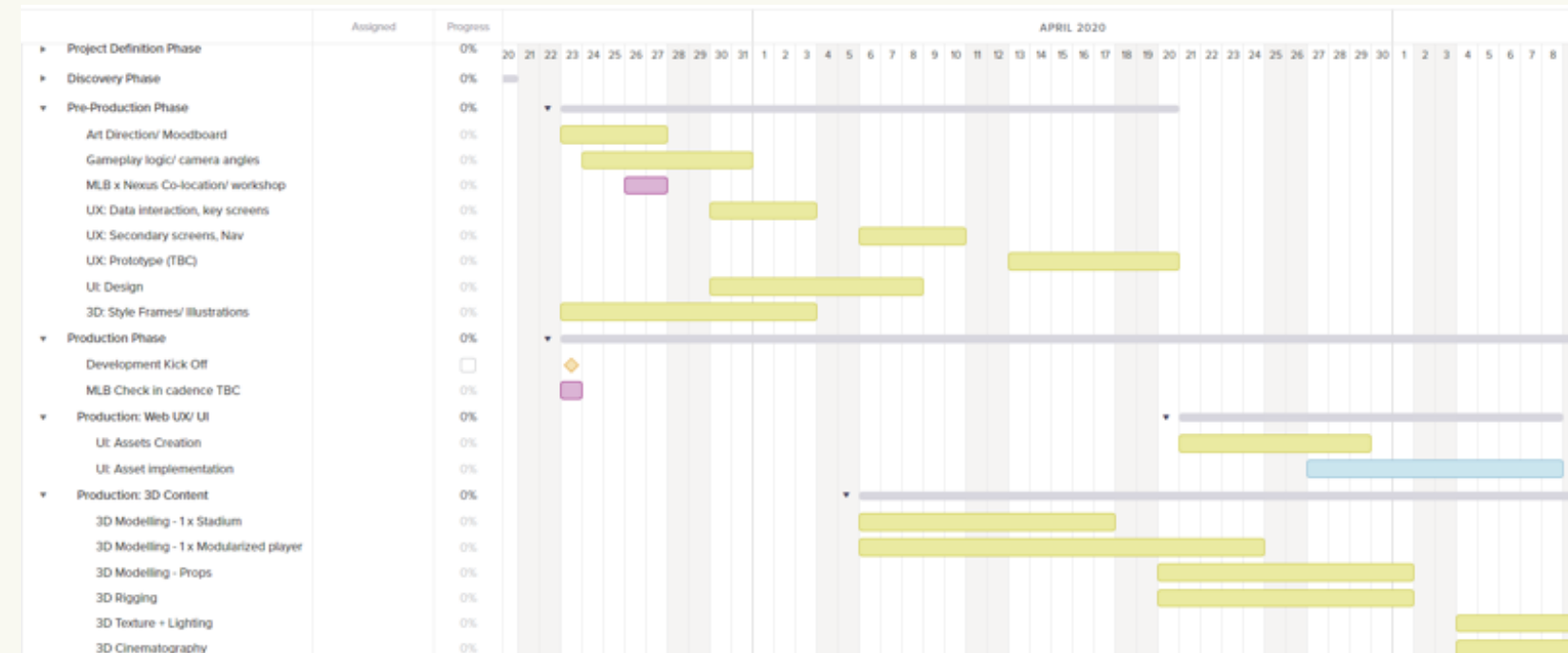
- Third-Party Development / Work-for-Hire
- Licensed IP & Brands
- Firm/fixed ship dates / day-and-date releases
- Firm/fixed budgets

The Challenge

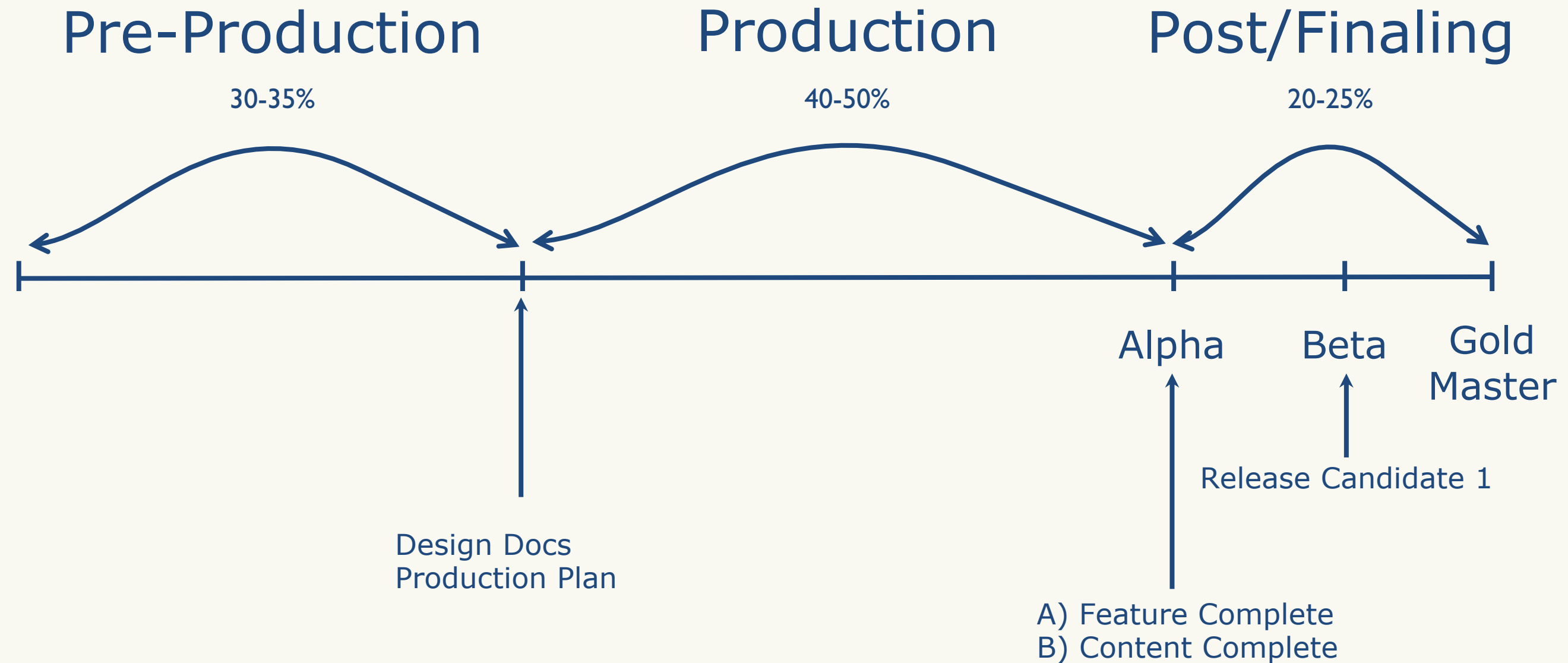
Deliver projects that meet their contractual obligations at the highest possible level of quality, on-time, and on-budget

Waterfall Production Methodology

- Originally established in appox. 1956
- Characterized by extensive Pre-Planning/Pre-Production
- Detailed Project Plans / Gantt Charts
- Critical Path analysis
- Popular for construction projects and physical engineering



Waterfall Production Methodology



Waterfall Production Methodology

Strengths

- Predictability
- Content & Design Dependencies
- Documentation approval process reduces ambiguity
- Team Member departures are (theoretically) less disruptive

Weaknesses

- Documentation quickly becomes outdated
- Ill-suited for vaguely-defined Projects
- Major changes to design/vision can be extremely disruptive and difficult to accommodate
- Impossible to accurately predict what will be fun
- **Consistently low-quality games**

Waterfall Production Methodology

“It is argued that the waterfall model can be suited to projects where requirements and scope are fixed, the product itself is firm and stable, and the technology is clearly understood.” – Wikipedia

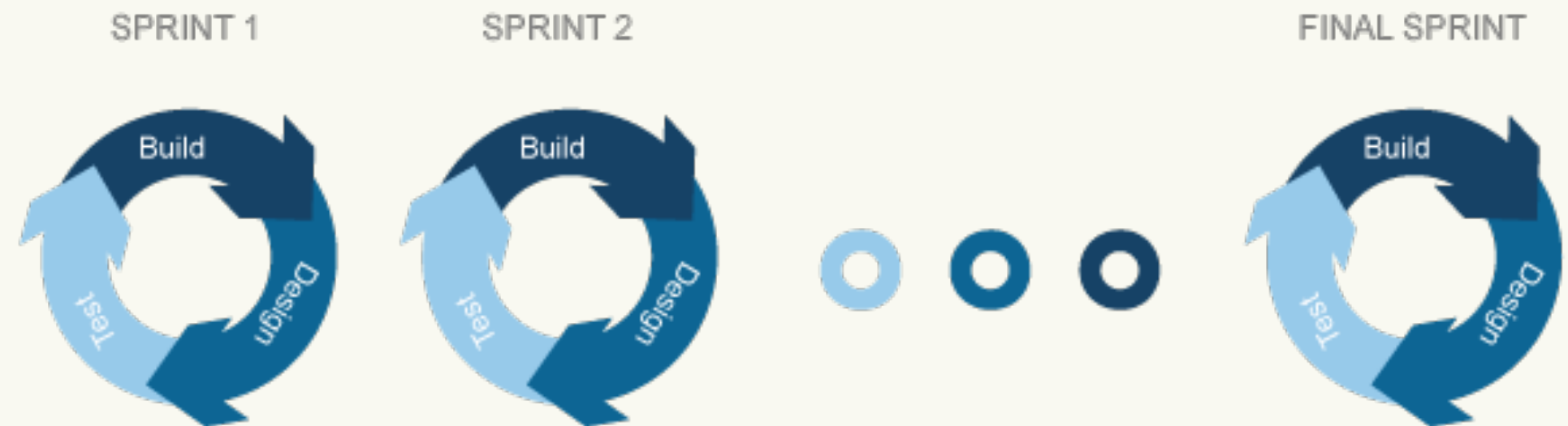
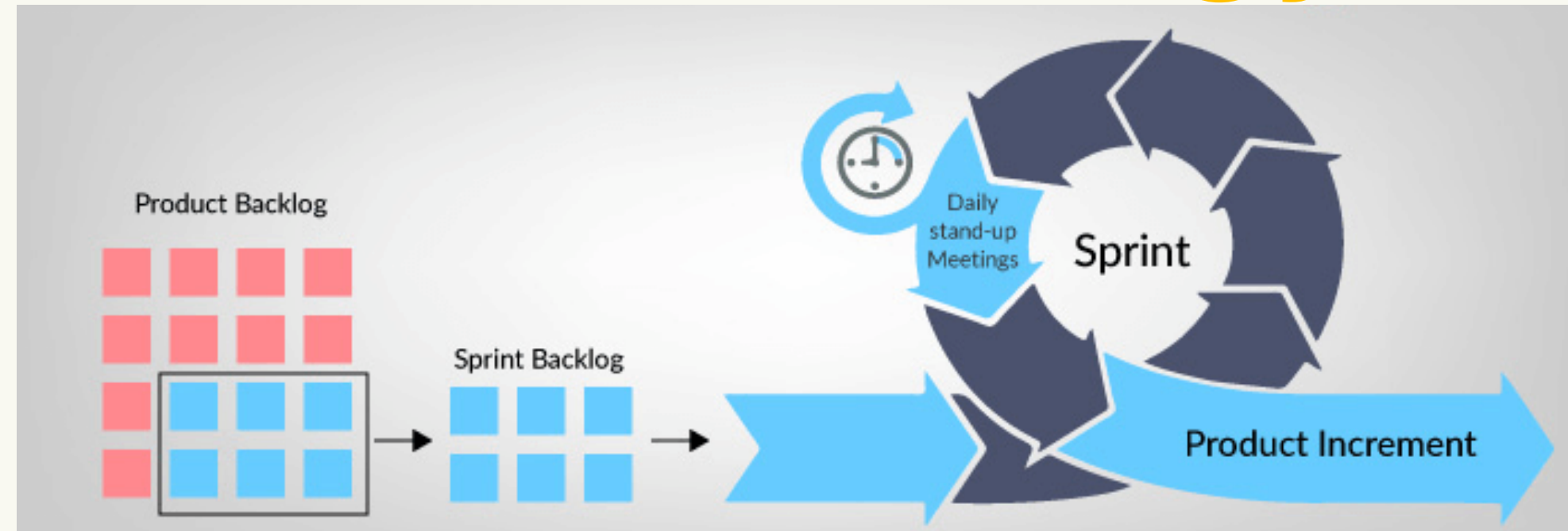
The Agile Manifesto



- “The Manifesto for Agile Software Development” was Published in 2001
- Was a reaction to the prevailing “waterfall” methodologies
- Based on the following values:
- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation*
- *Responding to change over following a plan*
- (Notably NOT conceived for video games)

Agile Production Methodology

- Backlogs
- Sprints
- Continuous Testing
- Rapid Iteration
- Velocity



Agile Production Methodology

Strengths

- Flexibility
- Quality
- Less-Defined Projects

Weaknesses

- Less Predictable
- Team member departures very disruptive due to lack of documentation
- Content Dependencies
- Prone to poor use of time & resources
- Very hard to see schedule risk coming
- **Consistently late, incomplete, or over-budget**

Conclusions

Agile

- Good: High Quality Work Product
 - Live Products / Games As Service
 - Small Projects
 - Polishing
- Bad: Inflexible Commitments
 - Contractual Scope Obligations
 - Fixed Ship/Delivery Dates
 - Fixed Budgets
 - Content-Dependent Designs

Waterfall

- Good: Meet contractual obligations
 - Deliver on-time
 - Deliver on-budget
 - Deliver what was designed
- Bad: Make fun games
 - No time for Polish
 - Very difficult to design “fun”

The Problem

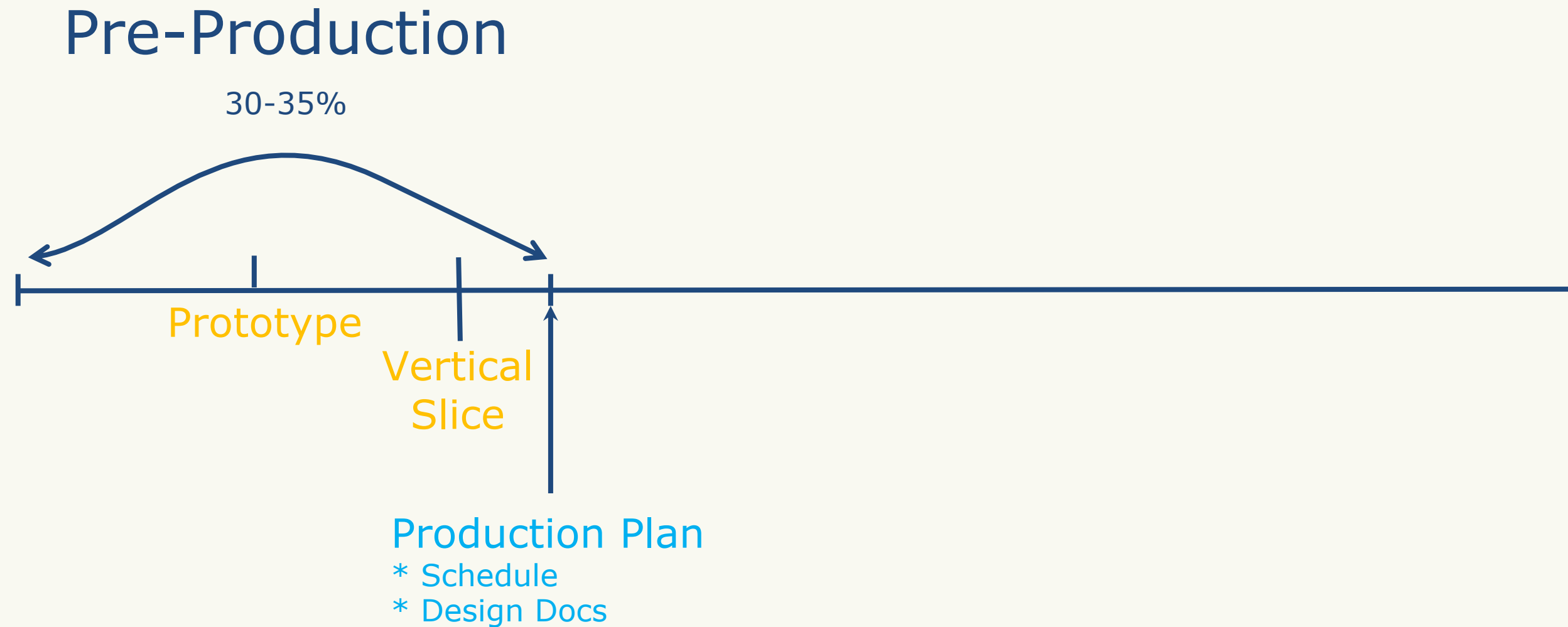
Despite both methodologies having existed for many years, many games still fail due to low quality, running out of time, or money.

The Solution

- A Hybrid Game Development Methodology
- Leverage the strengths and avoid the weaknesses of both methodologies
- There is no one-size-fits-all formula
- Be flexible/adapt to the project



A Hybrid Game Production Methodology



Pre-Production (Agile & Waterfall)

Prototype

- Core Mechanic / Fun Factor
- Key Questions
- Major Risks

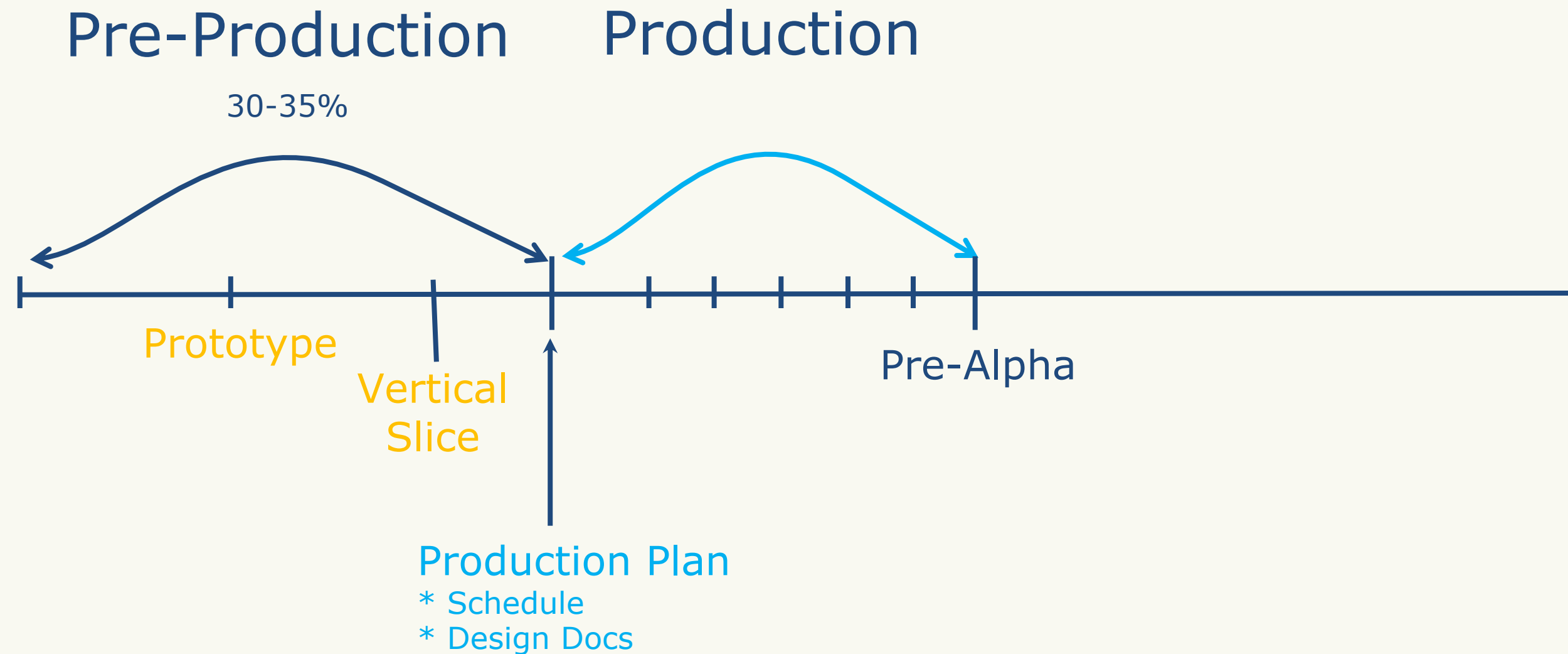
Vertical Slice

- Visual Bar
- Major Asset Pipelines
- Useful for stakeholders

Production Plan

- Game Design Doc (GDD)
- Technical Design Doc (TDD)
- Style Guide (VDD?)
- Deliverables Lists
- Asset Counts/Lists
- Production Schedule

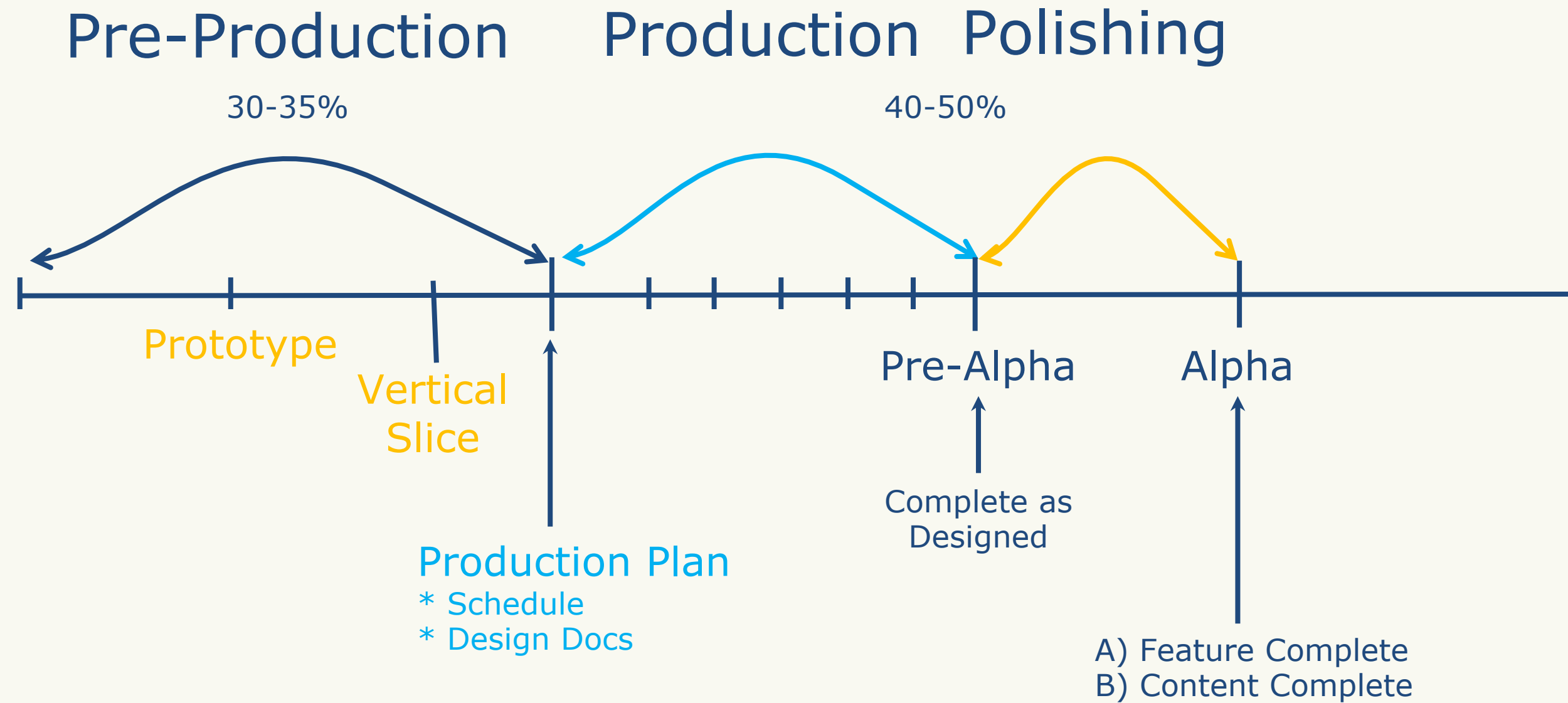
A Hybrid Game Production Methodology



Production (Waterfall)

- Production Milestones*
 - Themes
 - Payments
- Pre-Alpha ("Complete As Designed"/MVP)
 - DO NOT SLIP
 - Setup issue-tracking system
 - Polish Phase Backlog & Prioritization

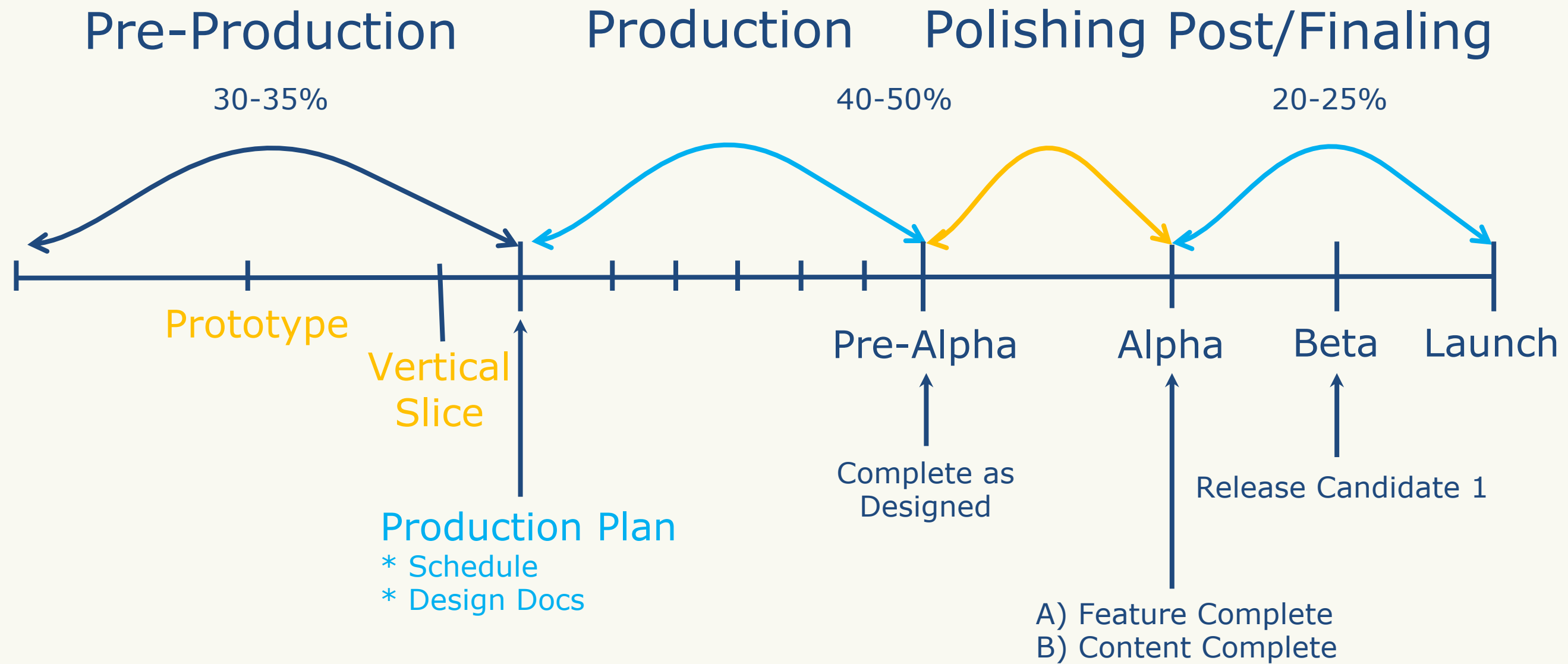
A Hybrid Game Production Methodology



Polishing (Agile)

- Alpha (Feature & Content Complete)
 - Polish Phase (Found Work & Rework)
 - Final Audio (especially VO)
 - Tutorial/FTUE
 - QA/Test Plan
 - Live Ops Backlog & Prioritization

A Hybrid Game Production Methodology



Post-Production (Waterfall)

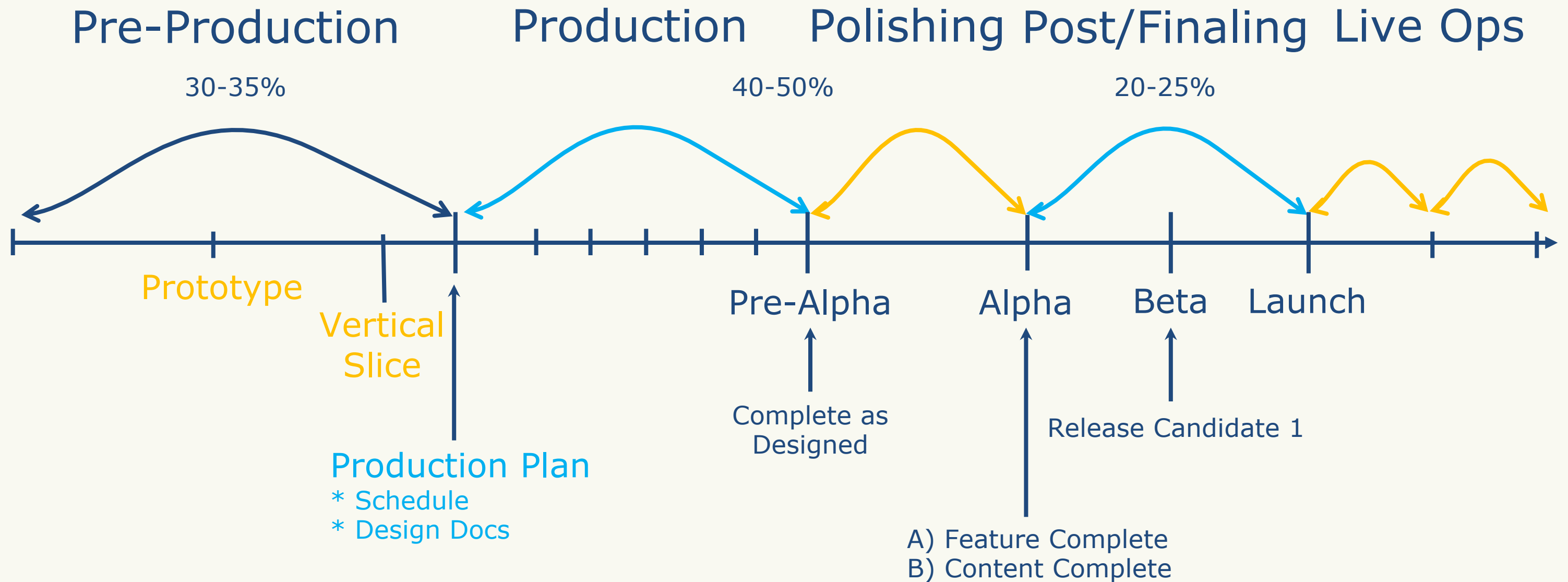
Beta/ZBR/RC-1

- Performance Optimization *
- Content QA & Debugging
- Functionality QA & Debugging

GM/Release

- Localization QA & Debugging
- Compatibility QA & Debugging
- Compliance QA & Debugging

A Hybrid Game Production Methodology

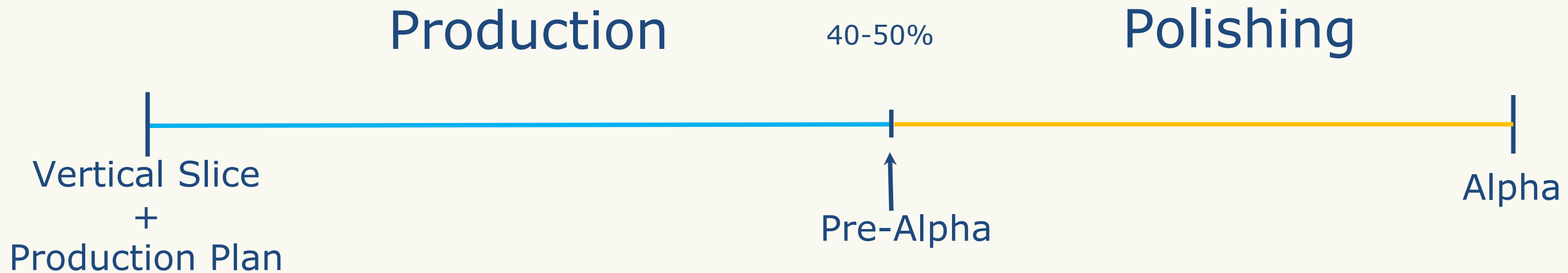


Live Operations (**Agile**)

- Naturally Iterative
- Based on User/Community Feedback & Metrics

The Pre-Alpha Slider

- Let's zoom-in on the Production & Polishing Phases
- Production + Polishing = ~40-50%
- What % of the project should be allocated for each?!



The Pre-Alpha Slider

Production

40-50%

Polishing

Indie/Solo
Self-Funded

Vertical Slice
+
Production Plan

Alpha

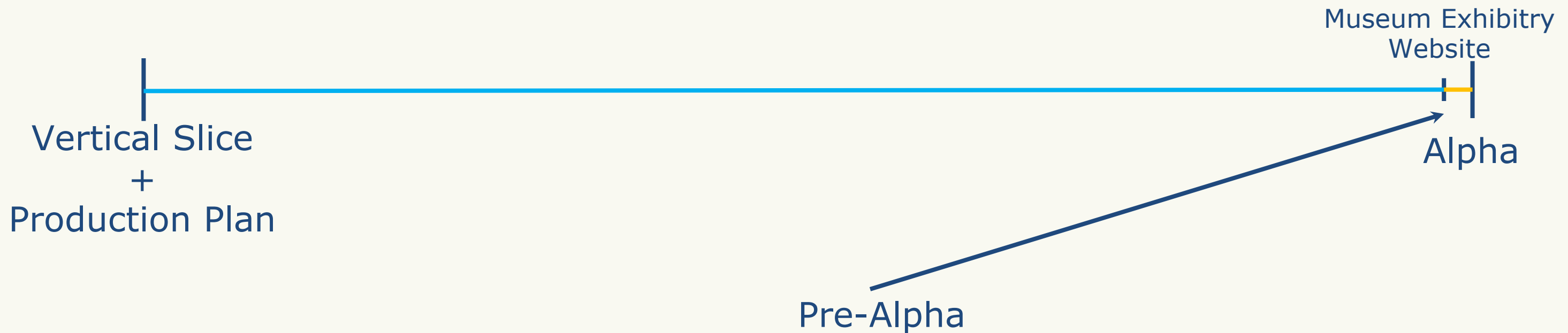
Pre-Alpha

The Pre-Alpha Slider

Production

40-50%

Polishing



The Pre-Alpha Slider

Production

40-50%

Polishing

Competitive Multiplayer
Narrative-Light
Tuning/Balance-Heavy

Vertical Slice
+
Production Plan

Alpha

Pre-Alpha

The Pre-Alpha Slider

Production

40-50%

Polishing

Mobile MVP
Annual Sequel
Narrative-Heavy

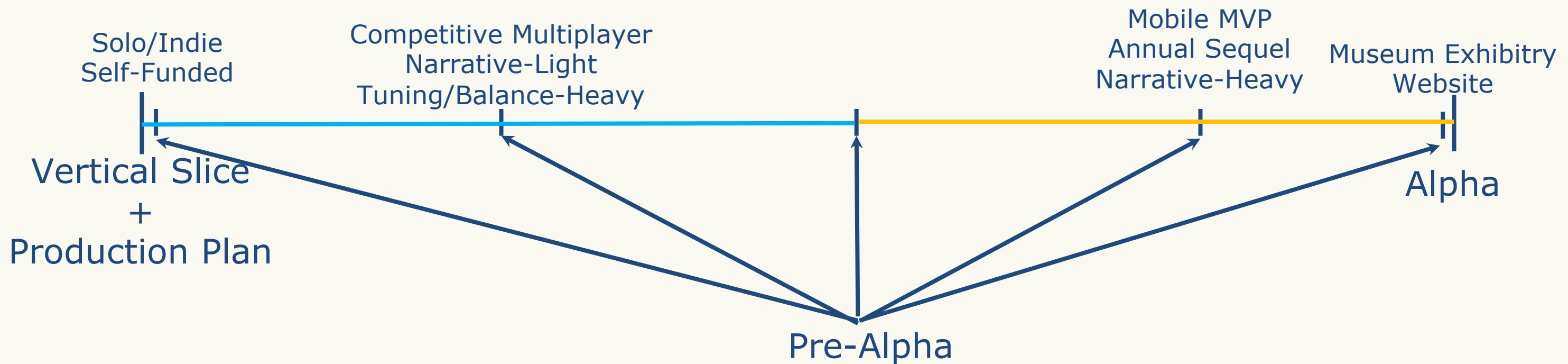
Vertical Slice
+
Production Plan

Alpha

Pre-Alpha

The Pre-Alpha Slider

- Every Project is Different
 - The trick is choosing where to put Pre-Alpha
- Production 40-50% Polishing



Summary

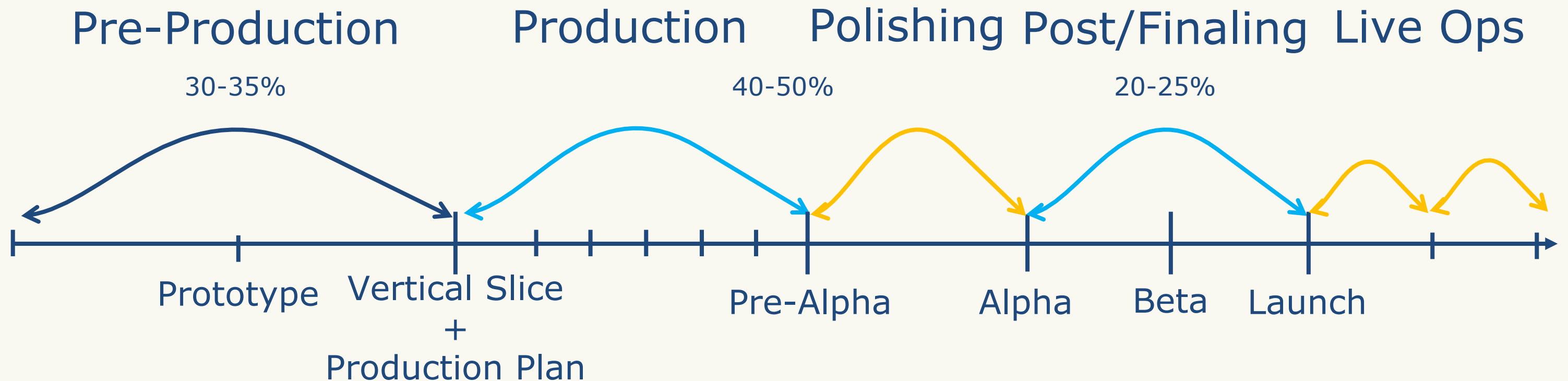
Agile

- Good: High Quality Work Product
- Bad: Inflexible Commitments

Waterfall

- Good: Meet contractual obligations
- Bad: Make fun games

A Hybrid Game Production Methodology



The Pre-Alpha Slider

Q&A

Please submit online feedback forms!

Speaker Wrap-Up Rooms:

South Hall, Room 204, Alcove 305 and Alcove 308

Appendix/References

- [Waterfall Model](#)
- [Agile Software Development](#)
- [Dynamic Systems Development Method \(DSDM\)](#)
- [Rational Unified Process](#)

The Challenge

- Game Development is the ultimate creative medium, drawing from ALL others:

Design	Art	Audio	Programming	QA
•Screen/Script writing	•2D Art & Animation	•Music	•Front-End Development	•Functionality
•Game Design	•3D Art & Animation	•Voice Acting (VO)	•Back-End Development	•Compatibility
•UX Design	•VFX	•Sound Design (SFX)	•Tools & Engine Development	•Compliance
	•Cinematography			•Localization

The Game Producer's Challenge: Create a plan and manage a team, drawing from all of these disciplines and deliver a project as designed, on-time, on-budget, and at an acceptable level of quality.

The 4 Levers

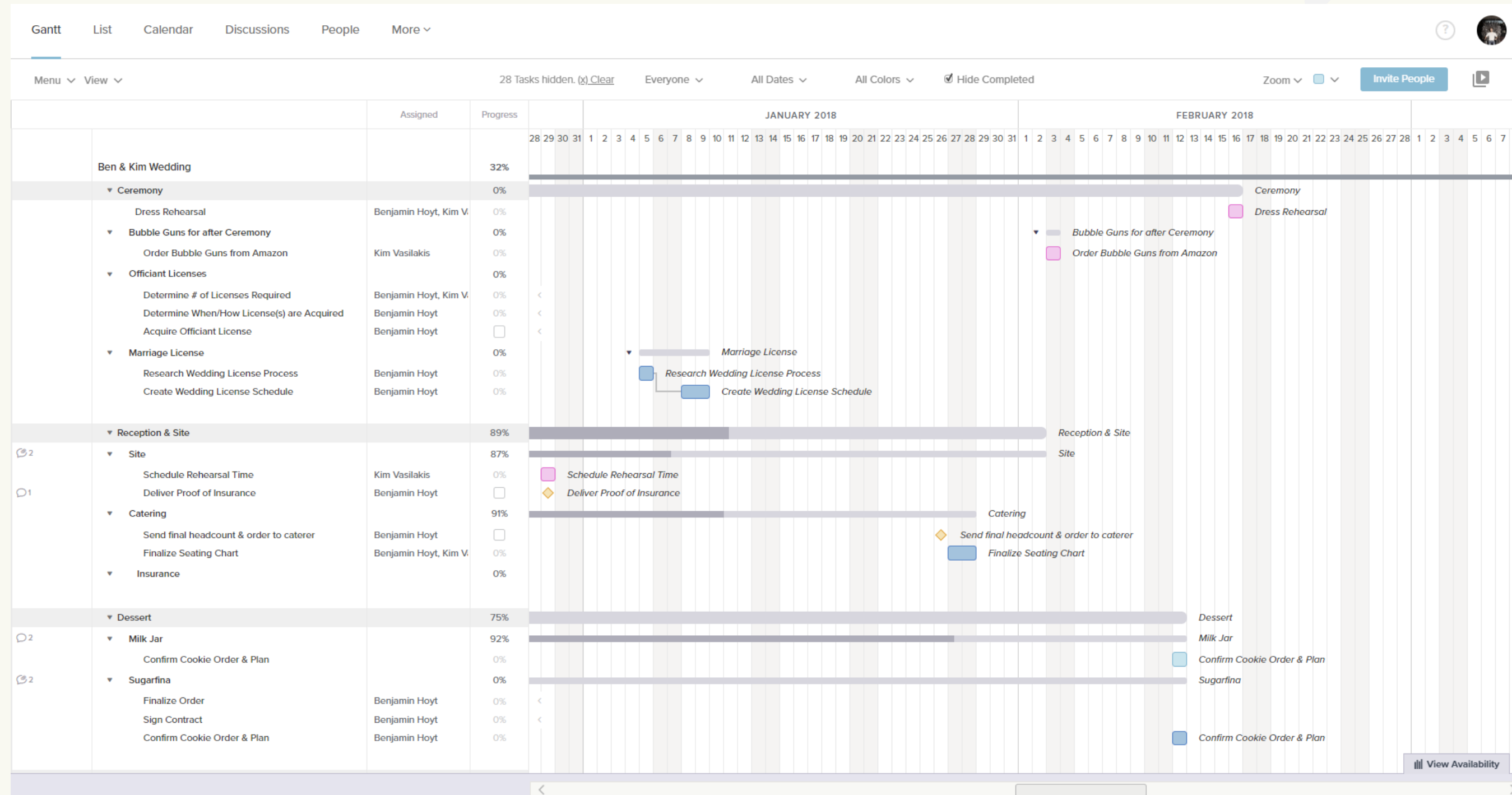
- Quality
- Budget
- Schedule
- Scope

What are your priorities?

Pick 2!

Be prepared to compromise on the others.

Don't Get Carried Away



Deliverable-Based Scheduling

- Define Deliverables
- Define Pipelines
- Derive Tasks*†‡
- Estimate Tasks
- Allocate Resources**
- Adjust Scope
- Create Gantt Chart