involutionstudios

Design directions for Agile

Tool-based vs. Task-based Interfaces

involutionstudios

	Tool-based Interface	Task-based Interface
Pros:		
Cons:		
Examples:		

Tool-based vs. Task-based Interfaces

involutionstudios

	Tool-based Interface	Task-based Interface
Pros:	Allows users to create their own workflow and processes. Implementation allows for scalable functionality at a broader tool level. Users can take the product to places not planned by the designers or developers.	
Cons:		
Examples:		

Tool-based vs. Task-based Interfaces

involutionstudios

	Tool-based Interface	Task-based Interface
Pros:	Allows users to create their own workflow and processes. Implementation allows for scalable functionality at a broader tool level. Users can take the product to places not planned by the designers or developers.	Optimized interface for specific tasks aimed at speed and repetition. Implementation allows for modular functionality at a more defined level. When specific in scope, more straight-forward to learn and use.
Cons:		
Examples:		

	Tool-based Interface	Task-based Interface
Pros:	Allows users to create their own workflow and processes. Implementation allows for scalable functionality at a broader tool level. Users can take the product to places not planned by the designers or developers.	Optimized interface for specific tasks aimed at speed and repetition. Implementation allows for modular functionality at a more defined level. When specific in scope, more straight-forward to learn and use.
Cons:	Generally more time intensive to learn how the tools work. Requires users to develop their own strategies to solve specific tasks. If not policed, can become a toolset that is quite large to manage on all fronts.	
Examples:		

	Tool-based Interface	Task-based Interface
Pros:	Allows users to create their own workflow and processes. Implementation allows for scalable functionality at a broader tool level. Users can take the product to places not planned by the designers or developers.	Optimized interface for specific tasks aimed at speed and repetition. Implementation allows for modular functionality at a more defined level. When specific in scope, more straight-forward to learn and use.
Cons:	Generally more time intensive to learn how the tools work. Requires users to develop their own strategies to solve specific tasks. If not policed, can become a toolset that is quite large to manage on all fronts.	More difficult for users to architect their own workflow or processes. More difficult to scale as a suite, even if scalability within a module works. Less integration across multiple sets of modules.
Examples:		

	Tool-based Interface	Task-based Interface
Pros:	Allows users to create their own workflow and processes. Implementation allows for scalable functionality at a broader tool level. Users can take the product to places not planned by the designers or developers.	Optimized interface for specific tasks aimed at speed and repetition. Implementation allows for modular functionality at a more defined level. When specific in scope, more straight-forward to learn and use.
Cons:	Generally more time intensive to learn how the tools work. Requires users to develop their own strategies to solve specific tasks. If not policed, can become a toolset that is quite large to manage on all fronts.	More difficult for users to architect their own workflow or processes. More difficult to scale as a suite, even if scalability within a module works. Less integration across multiple sets of modules.
Examples:	Adobe Photoshop, Microsoft Outlook, Netscape Navigator	Windows MediaPlayer, iTunes Music Store, Google Maps **Note:** Wizards are a subset of task-based interfaces, but do not define the genre.

Deep Dive approach

Examining a contained set of functionality in the product line with the goal of creating a strong solution to solve a specific problem. The solution, when well executed, can be applied to other aspects of the product line for future development.

Deep Dive approach

Examining a contained set of functionality in the product line with the goal of creating a strong solution to solve a specific problem. The solution, when well executed, can be applied to other aspects of the product line for future development.

Pros:

- Easy win: if we get the module right, everyone is excited.
- Contained design problem; allows for a sharp design focus.
- Sets up structure to fix all the other modules.

Deep Dive approach

Examining a contained set of functionality in the product line with the goal of creating a strong solution to solve a specific problem. The solution, when well executed, can be applied to other aspects of the product line for future development.

Pros:

- Easy win: if we get the module right, everyone is excited.
- Contained design problem; allows for a sharp design focus.
- Sets up structure to fix all the other modules.

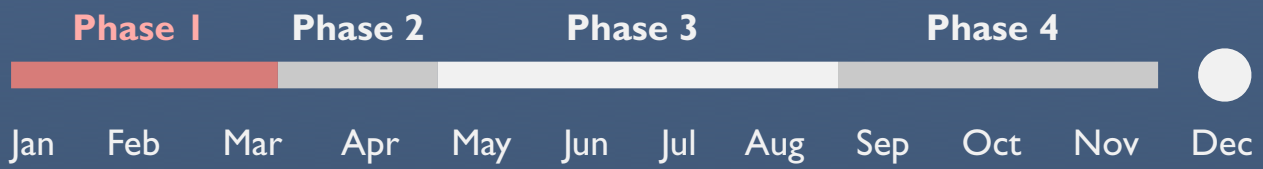
Cons:

- Limited scope: other products will clearly lag in comparison.
- Need to pick the right module for the project to be a success; very important.
- Inconsistency between products will require additional user training.
- As a suite, the products won't feel as integrated.

Deep Dive approach

Examining a contained set of functionality in the product line with the goal of creating a strong solution to solve a specific problem. The solution, when well executed, can be applied to other aspects of the product line for future development.

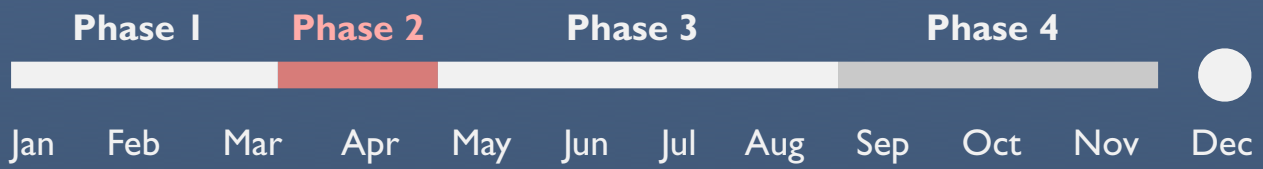
- Design strategy & blue sky period.
- Experiment with ideas and approaches on how to solve product problems.
- Architect task environments that are also scalable.
- Intensive front-loaded research on tasks for identification and validation.



Deep Dive approach

Examining a contained set of functionality in the product line with the goal of creating a strong solution to solve a specific problem. The solution, when well executed, can be applied to other aspects of the product line for future development.

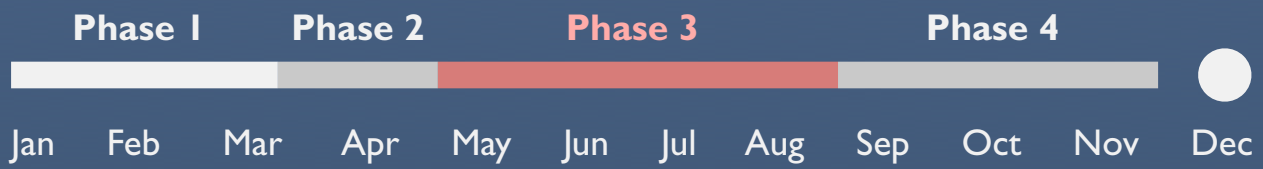
- Key design decisions made.
- Prototyping.



Deep Dive approach

Examining a contained set of functionality in the product line with the goal of creating a strong solution to solve a specific problem. The solution, when well executed, can be applied to other aspects of the product line for future development.

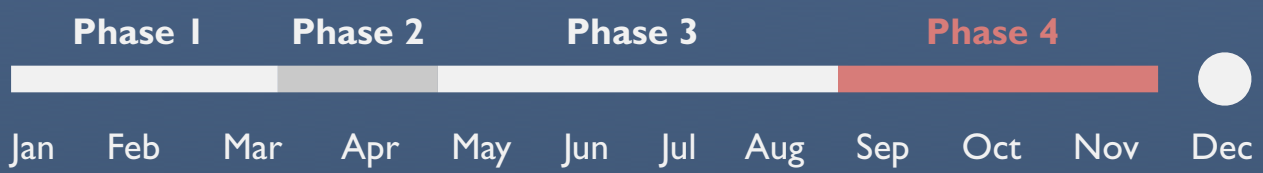
- Creating product with engineering team based on prototyping.
- Ongoing testing and feedback from alpha testers.



Deep Dive approach

Examining a contained set of functionality in the product line with the goal of creating a strong solution to solve a specific problem. The solution, when well executed, can be applied to other aspects of the product line for future development.

- Testing & tweaking



Clean Sweep approach

An overhaul of the existing product line to create a strong foundation on which to build and scale the product line over a period of 5 or more years.

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Clean Sweep approach

An overhaul of the existing product line to create a strong foundation on which to build and scale the product line over a period of 5 or more years.

Pros:

- Consistency across entire product suite.
- Straight forward to use without major re-education.
- Will make individual module redesigns in subsequent years more efficient.
- Provides a foundation for designing task based interfaces in the future.
- Could provide better platform for I18N.

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Clean Sweep approach

An overhaul of the existing product line to create a strong foundation on which to build and scale the product line over a period of 5 or more years.

Pros:

- Consistency across entire product suite.
- Straight forward to use without major re-education.
- Will make individual module redesigns in subsequent years more efficient.
- Provides a foundation for designing task based interfaces in the future.
- Could provide better platform for I18N.

Cons:

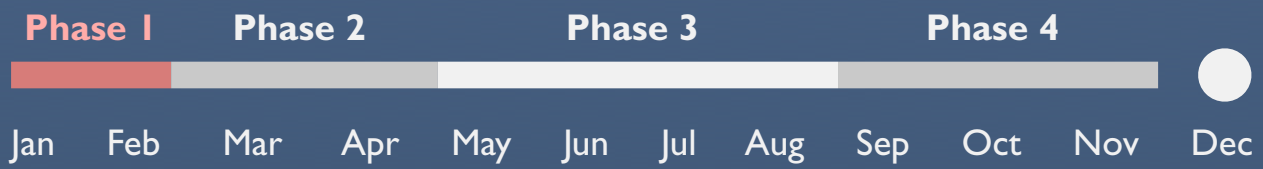
- Focus is on fixing what's there, not adding new features
- Touches everything: Will require a large amount of engineering and design work.
- Potentially risky to meet deadlines due to scope of the project.

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Clean Sweep approach

An overhaul of the existing product line to create a strong foundation on which to build and scale the product line over a period of 5 or more years.

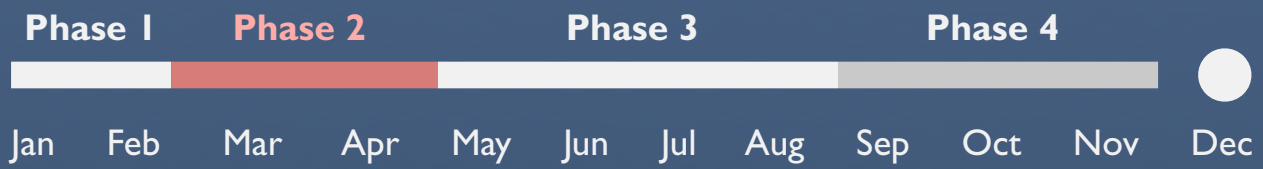
- Inventory of existing product interface and assets.
- Deep dive with engineers on architecture.
- Research to understand key features from the user's point of view.
- Identify product flow and pain points.
- Understand how people think about the system at the object and behavioral level.



Clean Sweep approach

An overhaul of the existing product line to create a strong foundation on which to build and scale the product line over a period of 5 or more years.

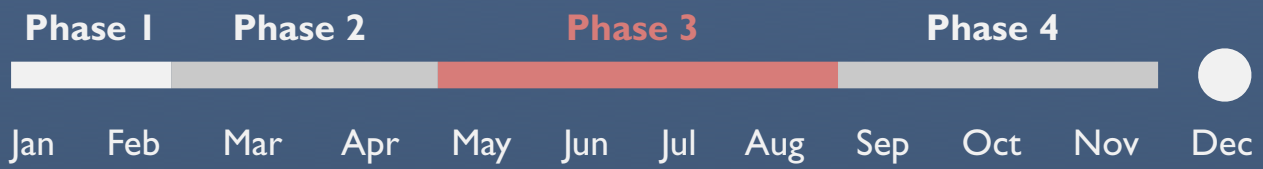
- Design and prototyping.
- Design decisions on structure, look and feel, programming & prototyping.
- Start getting key customers in and involved.



Clean Sweep approach

An overhaul of the existing product line to create a strong foundation on which to build and scale the product line over a period of 5 or more years.

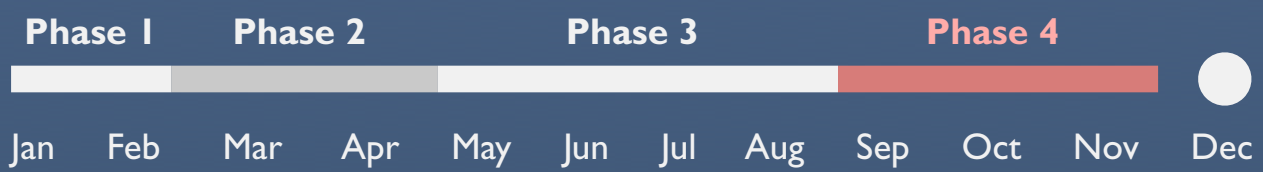
- Ripping everything out, redesigning it and putting it back together with engineering team.
- Ongoing testing and feedback from alpha testers.



Clean Sweep approach

An overhaul of the existing product line to create a strong foundation on which to build and scale the product line over a period of 5 or more years.

- Testing & tweaking.



involutionstudios

Design directions for Agile

Q&A