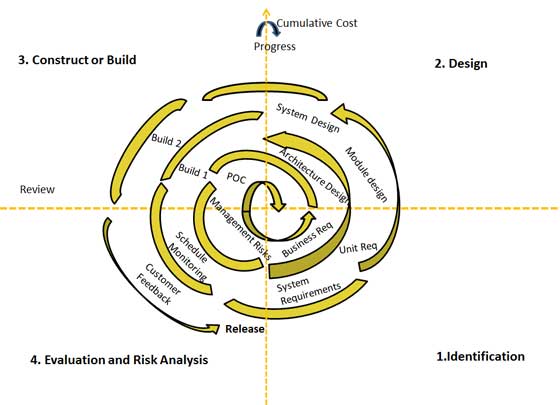
Harmony Betancourt

SID: 0003948694

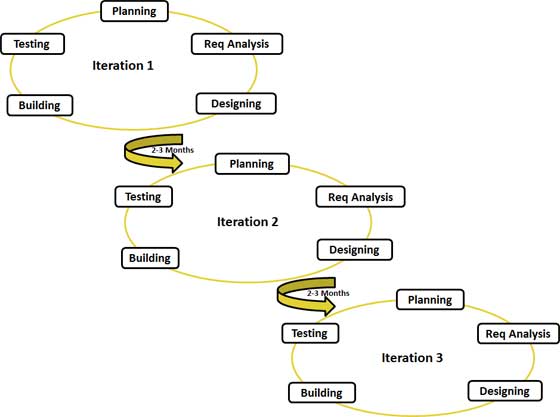
SDL Comparison

Spiral Model



|  |  |
| --- | --- |
| What: | Combines iterative development and waterfall model |
| Focus: | High emphasis on risk analysis |
| Phases: | Has four phases that are cycled through in a spiral formation, with iterations as needed through life of product   * Identification   + Base spiral  - requirements of job   + Subsequent spirals - system requirements - subsystem requirements - unit requirements - market * Design   + Base spiral  - concept   + Subsequent spirals  - architecture - logical modules - physical product - final product * Build   + Base spiral - proof of concept   + Subsequent spirals  - construct to meet requirements  - working models (with various version numbers) - adjust to customer feedback * Evaluation   + Base spiral - business risks   + Subsequent spirals - technical feasibility, cost, risks to company - customer feedback |
| Use: | Used often in projects with a set budget, high risk, long-term goals, uncertain customers, complex/unclear requirements, new products, or when significant changes are expected through the life of the software. |
| Pros | * Changing requirements are easily met * Prototypes are used * Customer feedback from an early stage * Risky development can begin early on |
| Cons | * Complex management * Date of finished project is unclear from onset * Complex process * Process may never end * A large amount of documentation is necessary |
| Recap | The Spiral Modal seemed daunting at first, but I find it to be the option I prefer between Agile and Spiral. With an ability to evaluate the risks and costs involved, the constant need for detailed documentation, and the flexibility needed to adapt, the Spiral Modal fits well with my own personal work style. The complexities involved in the process and management seem well worth it, as this modal is continually defining the expectations and requirements of every team member and the final product. While a team is very necessary to complete a long-term, high-risk software, the Spiral Modal emphasizes the individuals less and the team more. Having vast experience with teamwork, I will naturally gravitate to a system that can compensate for weaker team members, and allow for those on the team to easily pick up the work of one another. |

Agile Model



|  |  |
| --- | --- |
| What: | An adaptive incremental and iterative process |
| Focus: | Feature driven development with focus on adaptability |
| Phases: | Each iteration has five phases, with a build at the end to allow for customer and managerial feedback.   * Planning * Req Analysis * Designing * Building * Testing |
| Use: | Used with the following methodologies in mind: self-organization and motivation, working demos are more important than documentation, customer collaboration is required, quickly respond to change. |
| Pros | * Promotes teamwork * Functionally is quickly apparent * Low resource requirements * Works well with fixed or constantly changing requirements * Good for changing environments * Easily documented * Little or no planning needed to begin * Easy to manage |
| Cons | * Does not work well with complex dependencies * Greater risk * Requires an identified customer and project manager * Lack of documentation |
| Recap | The Agile Model is very popular, and it seems with good reason. Highly adaptive to the needs of the customer and client, this modal allows designers and developers to start immediately, and with little requirements for documentation. My main concern regarding this particular model is that it requires so little documentation, and depends so much on self-sufficient and motivated individuals and teamwork, that the loss of any team member could result in a substantial setback. |