|  |  |  |
| --- | --- | --- |
| ID | Notes | Source ID |
|  | **Extreme Programming Practices**  **Incremental planning:** Requirements are recorded on  Story Cards and the Stories to be included in a release are  determined by the time available and their relative priority.  The developers break these stories into development  "Tasks".  **Small Releases:** The minimal useful set of functionality  that provides business value is developed first. Releases of  the system are frequent and incrementally add  functionality to the first release.  **Collective Ownership:** The pairs of developers work on  all areas of the system, so that no islands of expertise  develop and all the developers own all the code. Anyone  can change anything.  **Sustainable pace:** Large amounts of over-time are not  considered acceptable as the net effect is often to reduce  code quality and medium term productivity.  **Continuous Integration:** As soon as work on a task is  complete, it is integrated into the whole system. After any  such integration, all the unit tests in the system must pass |  |
|  | Several successful large aerospace spiral projects were also discussed. The best documented  of these is the CCPDS-R project discussed in [Royce 98]. Its Ada Process Model was the  predecessor of the Rational Unified Process and USC MBASE approach, which have both  been used on a number of successful spiral projects [Jacobson 99, Boehm 98], as has the SPC  Evolutionary Spiral Process [SPC 94]. Further successful large aerospace spiral projects  were presented by SAIC and TRW [Kitaoka 00, Bostelaar 00]. |  |
|  | C-Bridge’s RAPID  Define, Design, Develop, and Deploy   * **Define** business requirements * **Design** an optimal approach and technical blueprint * **Develop** the solution * **Deploy** the solution | 4 |
|  | Refinements of the spiral model such as the Software Productivity Consortium's  (SPC) Evolutionary Spiral Process (SPC) [SPC 94] and Rational, Inc.'s Rational Unified Process  (RUP) [Royce 98, Kruchten 98, Jacobson 99]. |  |
|  | Phases overlap implying that there is no fixed finish between the start of one phase and the start of the next. It also implies that previous phases are revisited when further information is found. |  |
|  | In a variation of this model, the software products, which are produced at the end of each step (or series of steps), can go into production immediately as incremental releases. |  |
|  | **WinWin Spiral Model**  A primary difficulty in applying the spiral  model has been the lack of explicit process guidance in  determining these objectives, constraints, and alternatives.  The Win-Win Spiral Model [Boehm 94] uses the theory  W (win-win) approach [Boehm 89b] to converge on a  system's next-level objectives, constraints, and  alternatives. This Theory W approach involves identifying  the system's stakeholders and their win conditions, and  using negotiation processes to determine a mutually  satisfactory set of objectives, constraints, and alternatives  for the stakeholders. In particular, as illustrated in the  figure, the nine-step Theory W process translates into the  following spiral model extensions:  Solicit  suggestions from stakeholders, evaluate them with respect  to stakeholders' win conditions, synthesize and negotiate  candidate win-win alternatives, analyze, assess, resolve  win-lose or lose-lose risks, record commitments and areas  to be left flexible in the project's design record and life  cycle plans. |  |