**CMM LEVEL 1 → 2**

**Waterfall:**

Part of the Methodology

|  |  |  |
| --- | --- | --- |
| **Name** | **Quote** | **Description** |
| Program Design | Step 1 of the Waterfall method [1 p. 331]   1. Begin the design process with program designers, not analysts or programmers. 2. 3) Write an overview document that is understandable, informative and current. [1 p. 331] | Royce describes the importance of designing and documentation of requirements and how the final product should look before starting the project. |

Key Process Area

|  |  |  |
| --- | --- | --- |
| **Name** | **Quote** | **Justification** |
| Requirement Management | Level 2 changes. Requirement Management:  Here we tried to sharpen the focus on requirements management as seen from a software-engineering perspective while recognizing that the development and revision of requirements typically are not the responsibility of the software engineering group. [2 p. 20]  Goal 2: Software plans, products, and activities are kept consistent with the system requirements allocated to software. [3 p. L2 2] | In level-2 CMM requirements are reviewed by managers and other affiliated groups. The software completion is verified based on the requirement drawn at the beginning of the project. If any changes occur it is discussed with the affiliated group otherwise the project is kept consistency as much as possible with initial commitment. This is similar to the program design and requirement documentation in the waterfall. |

Help/Hinder

|  |  |
| --- | --- |
| **Help/Hinder** | **Rationale** |
| Help | Because of the effort of keeping the software plan, a consistent Process becomes easier to comprehend. And at level 2 requirements are managed and the processes are planned, performed, measured and controlled. Since we have requirement documents and overall design the final product will likely satisfy the requirements, standards, and objectives. [4] |

[1] W. Royce, "Managing the Development of Large Software Systems," *Proceedings of the IEEE WESCON*, pp. 328-338, Aug. 1970   
[Online] Available: [http://dl.acm.org/citation.cfm?id=41801](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/CS%20432%20PDFs/Royce_waterfall.pdf)

[2] M. Paulk et al., "Capability Mature Model for Software, Version 1.1," Chapter 1-3*IEEE Software*, vol. 10, no. 4, pp. 18-27, Jul. 1993.  
[Online] Available: [http://dx.doi.org.byui.idm.oclc.org/10.1109/52.219617](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/CS%20432%20PDFs/Capability%20Mature%20Model%20for%20Software.pdf)

[3] M. Paulk et al., "Key Practices of the Capability Mature ModelSM, Version 1.1, "*Technical Report CMU/SEI-93-TR-025*, Feb. 1993.  
[Online] Available: [http://dx.doi.org.byui.idm.oclc.org/10.1109/52.219617](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/08%20-%20Paulk%20-%20Key%20Practices%20of%20the%20CMM.pdf)

[4] “CMMI Maturity Levels,” *Tutorialspoint*. [Online]. Available: https://www.tutorialspoint.com/cmmi/cmmi-maturity-levels.htm. [Accessed: 02-Nov-2019].

**Spiral Model:**

Part of the Methodology

|  |  |  |
| --- | --- | --- |
| **Name** | **Quote** | **Description** |
| Testing | This model stipulated that software be developed in successive stages (operational plan, operational specifications, coding specifications, coding, parameter testing, assembly testing, shakedown, system evaluation). [1 p. 62] | Testing is a part of iteration in Spiral. It will be unique for each iteration. |

Key Process Area

|  |  |  |
| --- | --- | --- |
| **Name** | **Quote** | **Justification** |
| Software Quality Assurance | Goal 1: Software quality assurance activities are planned.  Goal 2: Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively. [2 p. L2 64] | CMM says there must be a specification written at the beginning of the project to compare against for knowing the quality of software developed.  The QA of level 2 is itself not similar to testing of the Spiral method even though testing in Spiral is the closest to testing in Level-2.  We develop test cases in level-2 in the beginning but the same is not the case in Spiral. |

Help/Hinder

|  |  |
| --- | --- |
| **Help/Hinder** | **Rationale** |
| Hinder | Even though QA is the closest practice we can relate to testing in Spiral. Still, it doesn’t fulfill the requirement of QA. The testing is not consistent thorough the project life, it changes in each iteration. We need something consistent for Level -2. It hinders. |

[1] B. Boehm, "A Spiral Model of Software Development and Enhancement," *IEEE*, xxx, pp. 61–72, May 1988,   
[Online] Available: [http://ieeexplore.ieee.org/xpls/abs\_all.jsp?arnumber=59&tag=1](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/CS%20432%20PDFs/A%20Spiral%20Model%20of%20Software%20Dev.pdf)

[2] M. Paulk et al., "Key Practices of the Capability Mature ModelSM, Version 1.1, "*Technical Report CMU/SEI-93-TR-025*, Feb. 1993.  
[Online] Available: [http://dx.doi.org.byui.idm.oclc.org/10.1109/52.219617](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/08%20-%20Paulk%20-%20Key%20Practices%20of%20the%20CMM.pdf)

**XP:**

Part of the Methodology

|  |  |  |
| --- | --- | --- |
| **Name** | **Quote** | **Description** |
| Planning  Iteration to the first release | runs the planning game practice; agree the smallest set of stories be completed.  produce a set of functional test cases that should run at the end of iterations. [1 pg. 100] | Focuses on small iteration, the requirement changes every iteration, no consistency in the planning, and minimum effort in the documentation. |

Key Process Area

|  |  |  |
| --- | --- | --- |
| **Name** | **Quote** | **Justification** |
| Software Project planning | The software planning process includes steps to estimate the size of the software work products and the resources needed, produce a schedule, identify and assess software risks, and negotiate commitments [2 pg. L2 11]  Goal 1: Software estimates are documented for use in planning and tracking the software project [2 pg. L2 12]  Goal 2: Software project activities and commitments are planned and documented. [2 pg. L2 12]  Goal 3: Software project activities and commitments are planned and documented. [2 pg. L2 12] | Estimation is done at the beginning of software development; software is developed according to the planning and is consistent. If any changes happen in the plan, then the affiliated groups are informed. This is similar to the planning in XP but is long term. |

Help/Hinder

|  |  |
| --- | --- |
| **Help/Hinder** | **Rationale** |
| Hinder | Level 2 and XP is totally opposite. One focuses on consistency and another in flexibility. There is no documentation in XP, but in Level 2 CMM, planning is done in the early stage for overall project as well as the requirements and test cases. |

[1] R. Juric, "Extreme Programming and its Development Practices" *22nd Int. Conf. Information Technology ITI 2000*, Jun. 2000   
[Online] Available: [http://ieeexplore.ieee.org/xpls/abs\_all.jsp?arnumber=915842&tag=1](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/CS%20432%20PDFs/Extreme%20programming%20and%20its%20development%20practices.pdf)

[2] M. Paulk et al., "Key Practices of the Capability Mature ModelSM, Version 1.1, "*Technical Report CMU/SEI-93-TR-025*, Feb. 1993.  
[Online] Available: [http://dx.doi.org.byui.idm.oclc.org/10.1109/52.219617](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/08%20-%20Paulk%20-%20Key%20Practices%20of%20the%20CMM.pdf)

**Scrum:**

Part of the Methodology

|  |  |  |
| --- | --- | --- |
| **Name** | **Quote** | **Description** |
| Product Backlog  Increments | Force-ranked (prioritized) list of desired functionalities. Visible to all stakeholders. [1]  The product capabilities completed during the Sprints. [1] | Features needed are prioritized by the product owner. The developer knows what is done and what still needs to be done.  Increment keeps the record that what features are developed in the software. |

Key Process Area

|  |  |  |
| --- | --- | --- |
| **Name** | **Quote** | **Justification** |
| Requirement Management  Software Project Tracking and Oversight | Level 2 changes. Requirement Management:  Here we tried to sharpen the focus on requirements management as seen from a software-engineering perspective while recognizing that the development and revision of requirements typically are not the responsibility of the software engineering group. [2 p. 20]  Software Project Tracking and Oversight involves tracking and reviewing the software accomplishments and results against documented estimates, commitments, and plans, and adjusting these plans based on the actual accomplishments and results [2 p. L2 31]  Goal 1: Actual results and performances are tracked against the software plans  [2 pg. 32] | Requirements in Level-2 are not the responsibility of programmers as well as in Scrum which is determined by product owner.  Software projects are tracked against their plans. We compare what we have planned to complete in a given time and money and what we have completed. It is basically comparing increment and initial product backlog. |

Help/Hinder

|  |  |
| --- | --- |
| **Help/Hinder** | **Rationale** |
| Help | In level-2 we want somewhat consistent requirements and the document to record our progress and feature completed. This can be achieved with the help of product backlog and increment, so it helps. |

[1] M. James, "Scrum Reference Card," [Online] Available:[http://scrumreferencecard.com/ScrumReferenceCard.pdf](https://content.byui.edu/items/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/?.vi=file&attachment.uuid=e38e0945-caed-4675-a65f-8eeb24eeb4ff)

[2] M. Paulk et al., "Key Practices of the Capability Mature ModelSM, Version 1.1, "*Technical Report CMU/SEI-93-TR-025*, Feb. 1993.  
[Online] Available: [http://dx.doi.org.byui.idm.oclc.org/10.1109/52.219617](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/08%20-%20Paulk%20-%20Key%20Practices%20of%20the%20CMM.pdf)

**Grade Claim:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Exceptional 100% | Good 90% | Acceptable 70% | Developing 50% | Missing 0% |
| Methodology and Process Area 40% | There is *no room for improvement* in describing the development methodology or the process area. | Both the CMM Level 2 process areas and the development methodologies are accurately defined. | There exists one small factual error or something is not fully defined. | There exists one large error, multiple small factual errors, or large parts not fully defined. | Large parts of the development methodology or CMM process are inaccurately described or not described at all. |
| Help/Hinder 40% | The rationale is convincing. | The rationale makes a strong case that the development methodology will help or hinder progress to Level 2. | One aspect of the rationale does not contribute to the case that the development methodology will help or hinder progress to Level 2. | The rationale is overly vague, does not appear to be related to the process area, or does not appear to be related to the development methodology. | No attempt was made to make a rationale. |
| Professionalism 20% | The paper is easy to read and ideas are clearly communicated. | Everything is properly cited, there are no grammar or spelling errors, and the writing style is "professional." | One instance of a spelling error, grammar error, incomplete citation, overly verbose, poor formatting, or poor writing. | A citation is missing where one is needed (plagiarism alert!). | Gross spelling/grammar errors or other aspects of the writing that make the paper difficult to read. |