Focus area of application

|  |  |
| --- | --- |
| **Criteria** | **Measurement** |
| Has to be either Agile or Waterfall | * + Yes or No   + Agile Projects tend to have a shorter life span   + Waterfall project tend to have a longer life span |
| Universal: usable for most organizations | * + General make up of model   + 1Yes (specific) – 5 (General) |
| Must have all phases the requirement/user phase. Design phase, implementation phase, testing phase and deployment and maintenance phase in the SSDLC model | * + Looking at the general model for SSDLC and comparing   + Yes or no question |

Implementation of model

|  |  |
| --- | --- |
| **Criteria** | **Measurement** |
| Is SSDLC compatible with the development of the secure software | * + Identify the software that's being developed function.   + 100% compatible: The software has all of its intended functions. From its intended normal function and its security functionality,   + 75% compatible: the model only satisfies part of the security requirements and the functional requirements   + 50%: the model only satisfies the regular functions of the software and not the security functions   + 49% or below: the model does no satisfy neither security or regular function or the general model requirement |
| Phase that focuses on new or continued development | * + Yes or no question   + Yes: It does focus on new and continued development   + No: It does not |
| The model’s preliminary/Initiation phases are effective for the software | * + Highly Effective: Security Requirements are specified and or use cases in predictive model. Use case is identified in adaptive model. Provides security training or workshops on project or education security and software engineering, lesson on vulnerabilities , etc.   + Moderately effective: has security requirements or use case but no form of training on the project requirements and basic software engendering, lesson on vulnerabilities and security   + Ineffective: Does not have requirements or use case or if security requirements are added into an adaptive model |
| Has to have a maintenance and operation phase | * + Must have a phase where the software is in its operation state and necessary steps are taken to keep the software functioning and to prevent security breaches and vulnerabilities to appear   + Yes   + No |

Security Enhancements/implementation

|  |  |
| --- | --- |
| **Criteria** | **Measurement** |
| Security certified standard outside organizations | * + The review of standard security regulations outside the organization (OWASP, GDPR)   + Whether they are applied or not applied   + Certified: Unable to penetrate or hack a system, has authentication or authorization services   + No Certified: No Authentication or authorization services, many Vulnerabilities, Able to penetrate a system |
| Is Security testing done in more than one phase of the SSDLC? | * + In most SSDLC, security testing is done in two phase, you see testing done in the implementation phase and the actual testing phase   + Yes: security testing is done in two phases   + No: Only one phase has security testing |
| Security is emphasized throughout all phases of the SSDLC | * + In every phase, there should be some security activity being done   + 100% security implemented. All phases have some sort of security activity   + 50%: 3 out of 4-6 phases has some sort of security activity |
| Tailing to organizational security regulations/policy | * + Organizational policy is reviewed   + Scale 1 to 5 how close does SSDLC requirements for software follow organizational policies and regulations 1 (not close) 5(very close) |

Security training and staff

|  |  |
| --- | --- |
| **Criteria** | **Measurements** |
| Security training (Basic and specific) The difficulty of implementing training. | * + Scale of 1-10 (1-3 is easy, 4-6 is moderate and 7-10 is harder   + The easier and accessible training is, the better. The staff needs to be trained on security and the SSDLC before the project starts |
| Implementation of proper security staff and stakeholders | * + Yes or no   + No: If the production of the software fails or the software is not secure and has security flaws, there was no proper or no security staff or stake holders implemented   + Yes: If the production of the software and it's security aspect is successful and the software has little to no security problems then it can be concluded that there were proper security staff and supervision. |

Results:

Focus area of model

* First Criteria: As of now, no model works when both Waterfall and Agile are combined
* Second Criteria: The model that more specific is the Jones & Rastogi model (Criteria still being modified)
* Third criteria: Three of the models have all the phases required for the SSDLC model, the only one that does not have all the phases is the SEI model.
* Overall, so far, the Microsoft and the Russell & Jones are more effective in this group of criteria

Implementation of Software

* First criteria: The Jones & Russell model has the compatibility of 100% as well as Microsoft's Trustworthy computing security Developmental Life-Cycle

SEI is at 75% compatibility because it does not have its maintenance phase which focuses on more of the security components after the software is developed

The Gary McGraw Touch Point model is at 0% compatibility because as of now, there are no evidence that a predictive and adaptive approach can be combined into one model. However, It can be used at a waterfall approach **or** agile approach. Over all, Microsoft and the Russell & Jones model are more effective in this criteria

* Second criteria: Two models are the leading two in this area, they are the touch point model (Because of the fact that it can work for Agile development which allows for continued development). The Jones & Russell model has a disposal phase that focus on new or continued development.
* Third criteria: 3 models are leading in this criteria as highly effective: Microsoft's, model, Russell & Jones’ model, and SEI model. The Gary Touch Point model doesn’t work because the Security requirements are in an agile model.
* Criteria 4: The Microsoft, Gary Touch Point, and the Russell & Jones model have an operation and maintenance phase. Which makes them more effective models for SSDLC and at maintaining and controlling security. The SEI model does not an operation and maintenance phase which makes it less effective model for SSDLC
* Overall, in this criteria, so far, the Gary Touch Point, Microsoft, and Russell & Jones model were more effective

Security Enhancements and Implementation

* Criteria 1: Security testing is done in more than one phase. All models in this criteria pass for this criteria
* Criteria 2: Security is emphasized throughout all phases of the model. All models pass for this criteria
* Over all, all models passed in these criteria

Security Training and Staff

* Criteria 1: security training difficulty: SEI has the highest scale of 8 because team members attend workshops on common vulnerabilities, they also trained in fundamental software engineering practices and it is also only done in the requirement phase. Microsoft's model is the second highest at the scale of 7 because it only involves concepts and no coding training but developers and programmers have to attend a different security class each year. The training stretches out to two phases: Requirement and design phases. Russell & Jones model is the third highest at the scale of 5 because the training is done in the operations and maintenance phase and is ongoing. The Gary model is the lowest with the scale of 2 because little attention is given to training, It is already assumed that developers and other team members no the material.
* Criteria 2: Security staff: three of the models pass this criteria except for The Gary Touch Point model which does not provide a security professional for the security functions of the SSDLC
* Overall, so far, the most effective model in these criteria is the SEI model and the Microsoft model