GAI NetConsult GmbH— Am Borsigturm 58 — 13507 Berlin — www.gai-netconsult.de — Tel: 030/417898-0

# Standards und Verfahren für mehr Sicherheit im Software-Entwicklungsprozess (Teil 2)

Wilfrid Kettler (GAI NetConsult GmbH)

Dezember 2010 Sonderdruck aus Security *Journal* #52

**Viele Publikationen und Fachberichte (auch aus unserem Hause) beschäftigen sich mit Risiken, Schwachstellen und Angriffsszenarien bei produktiv betriebenen Webanwendungen. Dabei wird stets auf bekannte oder auch neu auftretende Sicherheitsprobleme beim Betrieb hingewiesen sowie Maßnahmen zur Abwehr von Angriffen empfohlen. Die angeprangerten Schwachstellen sind jedoch an verschiedenen Stellen der Webanwendung tief verankert, die Fehler wurden bereits im Entwicklungsprozess gemacht, übersehen oder bekannte Angriffsmuster wurden einfach nicht beachtet. Daher ist es besonders wichtig, die nachhaltige Verbesserung der Qualität und des Sicherheitsbewusstseins Entwicklungsprozess zu verankern. Unsere Berichte über verschiedene Ansätze zu Vorgehensmodellen, Standards oder Best Practices zum Informationssicherheits-Management und dem sicheren Softwareentwicklungsprozess möchten wir zum Ende des Jahres um eine Darstellung von sog. Reifegradmodellen für den Softwareentwicklungsprozess ergänzen.**

Im Allgemeinen hat sich die Auffassung und Erkenntnis gefestigt, dass Sicherheitsprobleme vielfach, wenn nicht sogar überwiegend, auf Design- und Programmierfehler, auf Nachlässigkeiten und/oder allgemein auf Qualitätsmängel zurückzuführen sind. Entsprechend wird weltweit sowohl bei Herstellern und Entwicklungshäusern, als auch auf nationaler und internationaler Ebene versucht, durch Anwendung eigentlich bekannter und bewährter Verfahren zum Qualitätsmanagement den Entwicklungsprozess von Software zu verbessern.

Man hat sich schon früh – wenngleich mit bekanntermaßen geringem Erfolg – bemüht, Software-Entwicklung hinsichtlich des Qualitätsmanagements weitgehend zu industrialisieren und am Prozessverständnis fertigender Unternehmen auszurichten. Bei industriell gefertigten Produkten ist es heute kein Thema mehr – Qualität und professionelle Fertigungsmethoden mit integrierter Qualitätssicherung sind für die Unternehmen unverzichtbar. Allein der Wettbewerb zwingt diese zur Effizienzsteigerung bei gleichzeitiger Qualitätsverbesserung (von formalen Regelungen und Gesetzen, die dieses erzwingen, einmal ganz abgesehen).

## Wie alles begann – wieder einmal finden wir die Wurzeln beim Militär…

Nicht nur für die Privatwirtschaft gilt die große Herausforderung, dass (Software)Lieferungen selten fristgerecht, innerhalb des vorgegebenen Budgets und unter Einhaltung bestimmter Qualitätsanforderungen erfolgen. Auch im Öffentlichen Bereich, z.B. im Rüstungsbereich, sind nicht erst jetzt (Stichwort „Airbus A400M"), sondern schon in der Vergangenheit immer wieder Großprojekte u.a. an Softwareproblemen einzelner Komponenten gescheitert. So kursieren Informationen über nicht veröffentlichte Reviews des US-Verteidigungsministeriums (Department of Defence) über den erheblichen Zeitverzug, die massive Budget-Überschreitung und selbst die Einstellung von bedeutenden Projekten – die allesamt auf Softwareprobleme zurückzuführen waren [1].
Aus dem Umfeld der Informationssicherheit bei Softwareeinsatz in der Privatwirtschaft und (zugänglichen) Stellen Öffentlicher Verwaltung hören wir nicht selten Aussagen verantwortlicher Manager, wie z.B. „Lieber fehlerhaft, aber fristgerecht liefern…; um Fehler / Sicherheit kümmern wir uns später…". Für die Auftraggeber bedeutet dies i.d.R., dass sie ihre Softwareprodukte nicht wie bestellt bekommen und dadurch z.B. Zusatzkosten entstehen, weil die Entwicklung teurer war oder weil der von der neuen Software erwartete Nutzen nicht, oder erst später realisiert werden kann. Der Auftragnehmer hat im günstigsten Fall, wenn z.B. die Strafe nicht vertraglich festgelegt ist, nur einen Imageschaden. Das allein kann aber schon für ein Unternehmen gravierende Folge haben.
Dieselbe Attitüde hätte beim Militär ganz sicher andere Auswirkungen (Stichwort: „Smart Weapons"). Ob nun tatsächlich wegen der spezifischen Risiken von Softwaremängeln in Rüstungsgütern oder doch nur wegen derselben Aspekte von Zeit und Budget – wir wissen es nicht – wurde auf Initiative des amerikanischen Verteidigungsministeriums 1984 die Gründung des Software Engineering Institutes (SEI) veranlasst. Die Hauptaufgabe des SEI war es, die Software-Qualität in der amerikanischen Industrie nachhaltig zu verbessern. In diesem Zusammenhang tauchte das erste Mal der Aspekt auf, dass das beauftragende Ministerium die Möglichkeit haben sollte, nicht erst das fertige (Software) Produkt, sondern den Auftragnehmer selbst, dessen Verfahrens- und Produktionsweisen – dessen Reifegrad an Qualität- bewerten zu können. Es galt als Ziel, bereits vor der Lieferung Aussagen darüber treffen zu können, ob und in welcher Güte beauftragte Software tatsächlich ausgeliefert werden kann.
Hier liegen die Wurzeln des **Capability Maturity Model (CMM).** Wesentliches Merkmal bei CMM ist die Bereitstellung der Methoden zur Bewertung einer Softwareorganisation und die Be-

stimmung der Maßnahmen zu ihrer Verbesserung. Jedes einzelne Prozessgebiet kann einen Fähigkeitsgrad (Capability Level) erreichen, der den Grad der Etablierung und Beherrschung der Aufgaben beschreibt. Darüber hinaus gibt es sogenannte Reifegrade (Maturity Levels), die eine Priorisierung der Themen für die Prozessoptimierung darstellen. Dabei geht man davon aus, dass die Qualität eines Softwareproduktes von der Qualität der Entwicklungsprozesse abhängt, mit denen es hergestellt wird.
Das Bewertungsverfahren durch einen Fragebogen wird als „Assessment" bezeichnet. Dieser Fragebogen kann auf den in CMM festgelegten Fragen zur Bestimmung der Ebene der Organisation basieren. Eine Ebene wird durch mehrere Hauptkriterien und Schlüsselbereiche (Key Process Areas) definiert. Ein Hauptkriterium wird durch bestimmte Aspekte oder Key Practices bezeichnet. Diese schreiben vor, was getan werden muss, um das jeweilige Hauptkriterium zu erfüllen.

**Geschichtliche Entwicklung – von (Software)-CMM zu CMMI…**

| Jahr | Entwicklung der Versionen von CMM(I) |
|---|---|
| **1986** | Auf Initiative des US-Verteidigungsministeriums begann das Software Engineering Institute (SEI) an der Carnegie Mellon University/Pittsburgh mit der Entwicklung eines Systems zur Bewertung der Reife von Softwareprozessen |
| **1991** | Das Modell wurde als Capability Maturity Model 1.0 herausgegeben |
| **1993** | Überarbeitung des CMM und Bereitstellung der Version 1.1 |
| **1997** | Kurz vor der Verabschiedung von CMM 2.0 wurde diese Version wegen unzulänglicher Unterstützung der Integration verschiedener Aktivitäten und anderer konzeptioneller Schwachstellen vom US-Verteidigungsministerium zurückgezogen und das CMMI-Projekt gestartet. |
| **2000** | Veröffentlichung der Pilotversion 1.0 von CMMI, damals noch unter dem Namen Capability Maturity Model Integrated |
| **2002** | Freigabe von CMMI unter dem neuen Namen Capability Maturity Model Integration |
| **2003** | Auslaufen der Unterstützung des SEI für die Ursprungsversion CMM; 2005 liefen auch die Lizenzen der Assessmentleiter für CMM aus, d. h. seitdem gibt keine offiziellen CMM-Assessments mehr |
| **2006** | Veröffentlichung der neuen Version 1.2 des CMMI und einiger grundlegender Veränderungen. So wurde u. a. die neue Version in CMMI-DEV (CMMI for Development) umbenannt; sie unterstützt die Optimierung von Organisationen, die Software, Systeme oder Hardware entwickeln |
| **2007** | Freigabe der Version 1.2 des CMMI for Acquisition (CMMI ACQ); sie unterstützt die Optimierung von Organisationen, die Software, Systeme oder Hardware einkaufen, aber nicht selbst entwickeln |
| **2009** | Freigabe der Version 1.2 des CMMI for Services (CMMI SVC); sie unterstützt die Optimierung von Organisationen, die Dienstleistungen anbieten - wie z.B. Krankenhäuser, Logistik- oder Beratungsunternehmen |
| **2010** | Am 1. November 2010 ist die Version 1.3 des CMMI für alle drei o.g. Varianten („Constellations") erschienen. Sie hat zum Ziel, eine Einheitlichkeit der behandelten 16 Prozessgebiete zu erreichen und die Referenzmodelle insgesamt zu optimieren; diese sind nun „aus einem Guss" und lassen sich nahtlos / besser kombinieren. |

**Capability Maturity Model Integration**
Das Capability Maturity Model Integration (kurz CMMI) ist ein Reifegradmodell zur Beurteilung und Verbesserung der Qualität („Reife") von Produkt-Entwicklungsprozessen in Organisationen. Dabei werden die Stärken und Schwächen einer Produktentwicklung objektiv analysiert. So können Verbesserungsmaßnahmen bestimmt und in eine sinnvolle Reihenfolge gebracht werden. Primär ist CMMI ein Mittel, die Produktentwicklung zu verbessern. Sekundär ist die offizielle Überprüfung eines Reifegrades („Appraisal", =Zertifikat) eine in der Industrie de-facto anerkannte Auszeichnung.

CMMI ist ein Prozessmodell und definiert im Gegensatz zu einer konkreten Prozessbeschreibung lediglich Anforderungen an eine gute Produktentwicklung (das „Was"), aber keine konkreten Schritte (das „Wie"). Das primäre Ziel von CMMI ist es, eine kontinuierliche Prozessverbesserung (**KVP**) zu unterstützen, indem Anforderungen bzw. Kriterien von einer professionellen Produktentwicklungsorganisation definiert werden. Die Definition des Entwicklungsprozesses obliegt der Organisation und ist eine wichtige Teilaufgabe der Prozessverbesserung. Da CMMI keinen konkreten Entwicklungsprozess definiert, kann es auf sehr unterschiedliche Organisationen und Organisationsgrößen angewendet werden. So kann z. B. die Forderung, dass bei der Projektplanung eine Zustimmung der Projektbeteiligten (Stakeholder) zum Projektplan eingeholt werden muss, auf sehr unterschiedliche Art und Weise konkret in einer Organisation umgesetzt werden. Es gibt daher nicht „die eine" richtige Umsetzung von CMMI.

Eine besondere Eigenschaft von CMMI ist, dass es nicht nur die Entwicklungsprojekte an sich verwaltet, sondern auch die projektbezogenen Aufgaben der Organisation, wie z. B. Bereitstellung von Ressourcen oder Durchführung von Trainingsmaß-

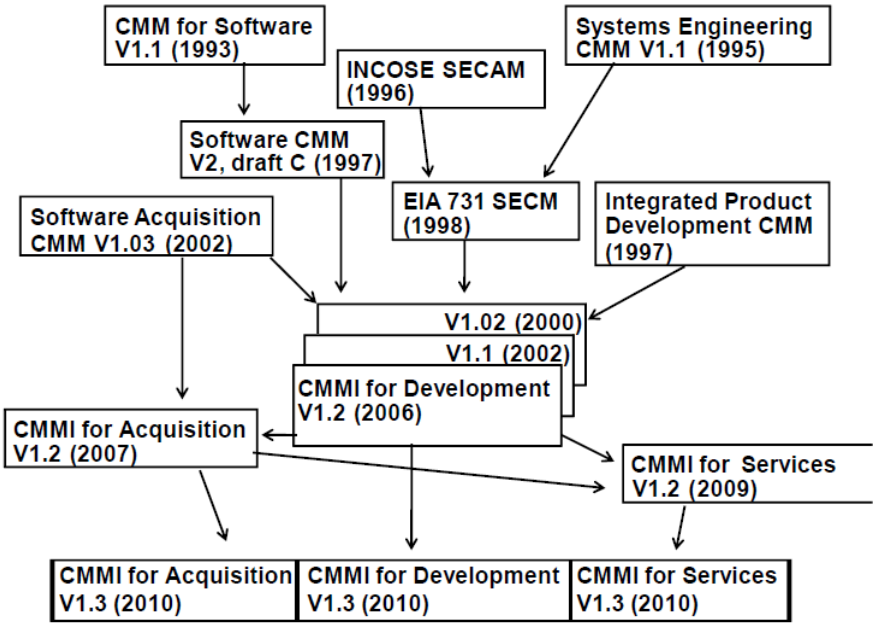


*Abbildung-1: Entwicklung der CMM(I)-Modelle*

nahmen, durchführt. Ein weiteres besonderes Merkmal ist, dass CMMI sehr viel Wert auf den gelebten Prozess legt und so im Gegensatz zu häufig als „Schrankware" bezeichneten Prozessen steht, die dokumentiert, aber nicht gelebt werden.

**Aufbau des Modells**

CMMI definiert eine Reihe von Prozessgebieten (z. B. Projektplanung, Anforderungsentwicklung, organisationsweite Prozessdefinition). Ein Prozessgebiet (Process Area) spezifiziert die Anforderungen an eine professionelle Produktentwicklung in einem bestimmten Gebiet durch ein Bündel verwandter Praktiken, die, sofern gemeinsam ausgeführt, eine Reihe von Zielen erfüllen, die für eine deutliche Verbesserung auf diesem Gebiet wichtig sind.

Die Prozessgebiete selbst sind in **vier Kategorien** eingeteilt:

- Projektmanagement (Project Management)
- Entwicklung (Engineering)
- Unterstützung (Support)
- Prozessmanagement (Process Management)

Während die ersten beiden Kategorien die Prozessgebiete enthalten, die typischerweise in Projekten umgesetzt werden, ist Prozessmanagement vor allem eine organisationsweite Aufgabe. Die Prozessgebiete in der Kategorie Unterstützung können sowohl eine Projektaufgabe als auch eine Organisationsaufgabe sein.

**Grundgerüst des Capability Maturity Models**

Die Grundlage eines kontinuierlichen Verbesserungsprozesses ist stets eine Anzahl von kleinen, evolutionären Schritten – keine revolutionäre Innovation. Die Schritte werden in CMM durch **fünf Reifeebenen** oder -stufen repräsentiert. Diese fünf Stufen definieren eine Skala für die Messung des Reifegrades des Softwareprozesses der Organisation und die Erhöhung der Qualität des Prozesses selbst. Die Reifeebenen stellen eine gut definierte Evolutionsgrundlage für das Erreichen der Reife eines Softwareprozesses dar. Jede Ebene schließt eine Anzahl der Ziele ein, bei deren Erreichen man eine wichtige Komponente des Softwareprozesses stabilisiert.

**Fähigkeitsgrade und Reifegrade**

CMMI geht die Verbesserung innerhalb eines Prozessgebiets durch so genannte **„Fähigkeitsgrade" (capability levels)** an. Ein Fähigkeitsgrad bezeichnet den Grad der Institutionalisierung eines einzelnen Prozessgebiets; der gesamte Prozess wird mit den Daten aus der statistischen Prozesskontrolle stetig verbessert und optimiert. Neben den Fähigkeitsgraden eines einzelnen Prozessgebiets definiert CMMI **„Reifegrade" (maturity levels).** Ein Reifegrad umfasst eine Menge von Prozessgebieten, die zu einem bestimmten Fähigkeitsgrad umgesetzt sein müssen. Jeder Reifegrad ist ein Entwicklungsplateau in der Prozessverbesserung der Organisation. Aufgrund des historischen Funda-

| Maturity Level | Merkmale |
|---|---|
| 1. Initial | Ad hoc; chaotische Prozesse kennzeichnen diesen Reifegrad; Erfolg ist von Personen und nicht von dem Einsatz bewährter Prozesse abhängig; Termin- und Budgetüberschreitungen sind üblich. |
| 2. Managed | Prozesse werden geplant durchgeführt, überprüft und gesteuert. |
| 3. Defined | Prozesse sind gut charakterisiert, werden verstanden und in Standards, Werkzeugen und Methoden beschrieben: Prozesse werden proaktiver und detaillierter als im 2. Level geplant und durchgeführt. |
| 4. Quantitatively Managed | Quantitative Zielvorgaben für Qualitätsüberprüfung und Prozessdurchführung sind etabliert, die Prozessdurchführung wird damit vorhersagbar. |
| 5. Optimizing | Es erfolgt eine kontinuierliche Prozessverbesserung aufgrund technologischer Innovationen und der Feststellung und Behebung allgemeiner Prozessstörungen. |

*Abbildung-2: Reifegrade ( Maturity Level)*

ments CMM, sind die Definitionen sehr ähnlich.

Die Bewertung des Reifegrades bzw. der Fähigkeitsgrade einer Organisation geschieht durch eine SCAMPI-Abschätzung (SCAMPI-Appraisal), die nur durch vom SEI autorisierte Personen geleitet werden kann. Die Liste aller vom SEI autorisierten Lead Appraiser, also diejenigen Personen, die ein solches SCAMPI leiten dürfen, findet sich auf den Seiten des Software Engineering Institutes. Die deutschsprachigen autorisierten Lead Appraiser haben sich im German CMMI Lead Appraiser and Instructor Board (CLIB) zusammengeschlossen.

**Weitere Prozessbeurteilungsverfahren**

Es existieren verschiedene Prozessbeurteilungsverfahren bzw. Modelle für das Qualitätsmanagement, die dem CMM(I) verwandt sind. Im Unterschied zur ISO 9001 gehen die CMMI-Modelle spezifisch auf die Praktiken in einem bestimmten Anwendungsgebiet ein. Während ISO 9001 die gesamte Organisation mit breiterer Betrachtung abdeckt, geht CMMI bei den konkreten Tätigkeiten weit mehr in die Tiefe und bietet konkrete Prozessgebiete und Praktiken. Die CMMI Modelle und die ISO 9001 haben jedoch denselben Grundgedanken, die Anforderungen der CMMI Modelle lassen sich so auch alle auf die Anforderungen ISO 9001 abbilden (siehe SEI, [2]).

Die CMMI Modelle setzen die Anforderungen von ISO/IEC 15504 (SPICE, Software Process Improvement and Capability Determination) an ein Prozessmodell um. Das Appraisal-Verfahren SCAMPI (Standard CMMI Appraisal Method for Process Improvement) setzt die Anforderungen der Norm ISO/IEC 15504 an ein Bewertungsverfahren teilweise um.
Neben den CMMI Modellen gibt es auch die Prozessmodellnormen ISO/IEC 12207 für Software - und ISO 15288 für die Systementwicklung. Im Gegensatz zu CMMI gehen diese beiden Normen aber weder über die Definition der Titel der Praktiken von CMMI hinaus, noch gibt es eine Integration beider Normen. Inhaltlich entsprechen ISO/IEC 12207 und ISO 15288 in etwa CMMI für Entwicklung (CMMI-DEV).

**Von der Theorie (zurück) zur Praxis…**

Der vorliegende Exkurs in die Herkunft und Definition der „Mutter der Reifegradmodelle" als akademische Grundlage ist z.B. wiedererkennbar in dem Microsoft Security Development Lifecycle (SDL), der im ersten Teil der Auseinandersetzung mit den Bemühungen der Softwareindustrie um eine bessere, sicherere Software, kurz dargestellt wurde. Microsoft SDL konzentriert sich dabei gleichermaßen sowohl auf die eigene Softwareproduktion als auch auf die Anwendbarkeit auf Dritte – jedoch mit den Microsoft-Entwicklungswerkzeugen. Unbenommen vom „ewigen Microsoft-Bashing" ist der Erfolg der konsequenten Umsetzung der Ideen des CMMI in den gesamten Softwareentwicklungsprozess nachweisbar und erkennbar.

Ebenfalls im ersten Teil dieses Artikels wurde BSIMM dargestellt – ein Reifegradmodell als Template zur Überprüfung und Anwendung im eigenen (Software) Unternehmen. Der bemerkenswerte Ansatz von BSIMM ist dabei, dass – im Prinzip als Feedbackschleife – erst einmal die Prozesse von einer Reihe (>30) von Unternehmen weltweit untersucht und erst danach die jeweils angestrebten und tatsächlich erreichten Prozessareas und deren Reifegrade klassifiziert wurden. Dennoch sind auch hier die Grundzüge dem CMMI entnommen.

**Software Assurance Maturity Model (SAMM)**

Gut fünf Jahre nachdem Microsoft den Security Development Lifecycle (SDL) eingeführt hat gibt es seit 2009 in der Version 1.0 das Opensource Pendant. Das Software Assurance Maturity Model [3] (kurz SAMM oder OpenSAMM) ist beim OWASP angesiedelt, bündelt die Konzepte und Diskussionen aus der OWASP-CMM-Projektgruppe und verfolgt ähnliche Ziele wie der Microsoft SDL:

**Ganzheitliche Integration der Sicherheit in die Softwareentwicklung**

Während dies bei Microsoft mit einem 14-teiligen Prozess umgesetzt wird, nutzt openSAMM ein Modell, welches auf vier sogenannten Business Funktionen beruht. Diese vier Business Funktionen werden mit jeweils drei Security Practices konkretisiert. Für jede Security Practice sind wiederum drei Reifegradebenen als zu erreichende Ziele definiert.

openSAMM ist ein offenes Framework, welches Unternehmen im Wesentlichen bei der Definition und Umsetzung einer Strategie zur Software-Sicherheit helfen und dabei die spezifischen Risiken des eigenen Unternehmens berücksichtigen soll. Konkret werden Anhaltspunkte und Unterstützung in folgenden Bereichen gegeben:

- Untersuchung der konkreten Maßnahmen zur Sicherheit von Software
- Aufbau eines ausgewogenen Programms zur Erreichung sicherer Software in definierten Schritten und Iterationen, konkreten Arbeitspaketen
- Metriken zur Darstellung konkreter Qualitätsverbesserungen der Security
- Definieren und Messen sicherheitsrelevanter Aktivitäten innerhalb einer Organisation

SAMM hat den Anspruch, so flexibel zu sein, dass es gleichermaßen von kleinen, mittleren und großen Unternehmen mit beliebigen Entwicklungsmethoden genutzt werden kann. Darüber hinaus soll dieses Modell wahlweise unternehmensweit oder aber auch nur für ein einzelnes Projekt angewendet werden können. Alle Projektphasen werden im Zusammenhang mit den vier

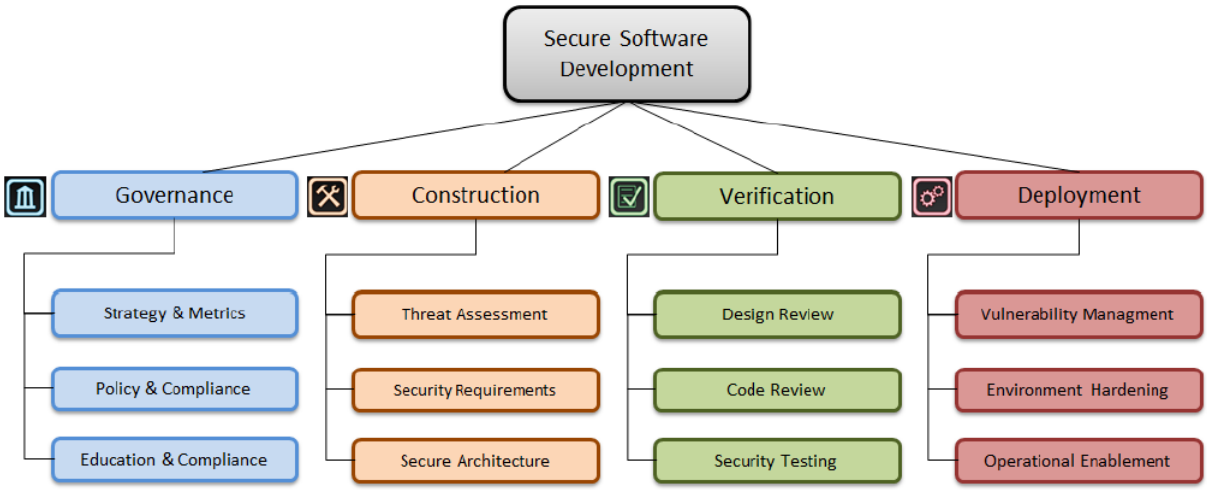


*Abbildung-3: Process-Areas und Kategorisierung von Maßnahmen*

wichtigsten Unternehmensbereichen abgebildet und können individuell höher bewertet / berücksichtigt werden – so z.B. der Bereich der Governance, als Voraussetzung für die erfolgreiche Umsetzung in einem Unternehmen. Folgende Abbildung zeigt die Anordnung einzelner Maßnahmenpakete entlang des Entwicklungsprozesses und bezogen auf die organisatorische Einheit des Unternehmens.

**Reifegradebenen**

Für jede der insgesamt 12 Security Practices sind 3 Reifegrad-Ebenen definiert; gemeinsamer Ausgangspunkt ist jeweils 0 – also „unreif". Im Detail gibt es bei den Reifegraden Unterschiede, doch generell finden folgende Punkte stets Berücksichtigung:

0: Startpunkt jeder Entwicklung, "unreifer" Zustand, keine Ziele erreicht
1: Grundverständnis / Bewusstsein ist vorhanden, Security Practices sind vereinzelt definiert
2: Effizienz und Wirksamkeit von Security-Practices wurden erkennbar gesteigert
3: Umfassende Beherrschung der Security Practices

**Analyse und Bewertung der tatsächlichen Reifegrade**

Ziel ist es, die internen Prozesse und Aktivitäten zur Sicherstellung der notwendigen Maßnahmen / Security Practices anhand der tatsächlich erreichten Ziele zu bewerten und hieraus wiederum Maßnahmen zur Verbesserung der Situation und zur Erhöhung des Reifegrades abzuleiten und letztlich umzusetzen.

Die tatsächliche Bewertung wird mit einem für das eigene Unternehmen passende bzw. realistisch erreichbare Referenzmuster verglichen und gilt somit als Zielvorgabe; die vollständige Erreichbarkeit höchster Reifegrade in allen Bereiche ist dabei eher unwahrscheinlich und kein konstruktives Ziel.

In der Tat scheint OpenSAMM viele Ähnlichkeiten mit dem Microsoft SDL zu haben - dies

| | Initiate | Define | Design | Develop | Test | Implement | Operate |
|---|---|---|---|---|---|---|---|
| Governance | Strategy & Metrics | | | | | | |
| | Policy & Compliance | | | | | | |
| | Education & Guidance | | | | | | |
| Construction | | Threat Assessment | | | | | |
| | Security Requirements | | | | | | |
| | | Secure Architecture | | | | | |
| Verification | | | Design Review | | | | |
| | | | | Code Review | | | |
| | | | | Security Testing | | | |
| Deployment | | | | | | | Vuln. Mgmt. |
| | | | | | | Environment Hardening | |
| | | | | | Operational Enablement | | |

*Abbildung-4: Security Practices entlang des Entwicklungsprozesses und bezogen auf die Unternehmenseinheit*

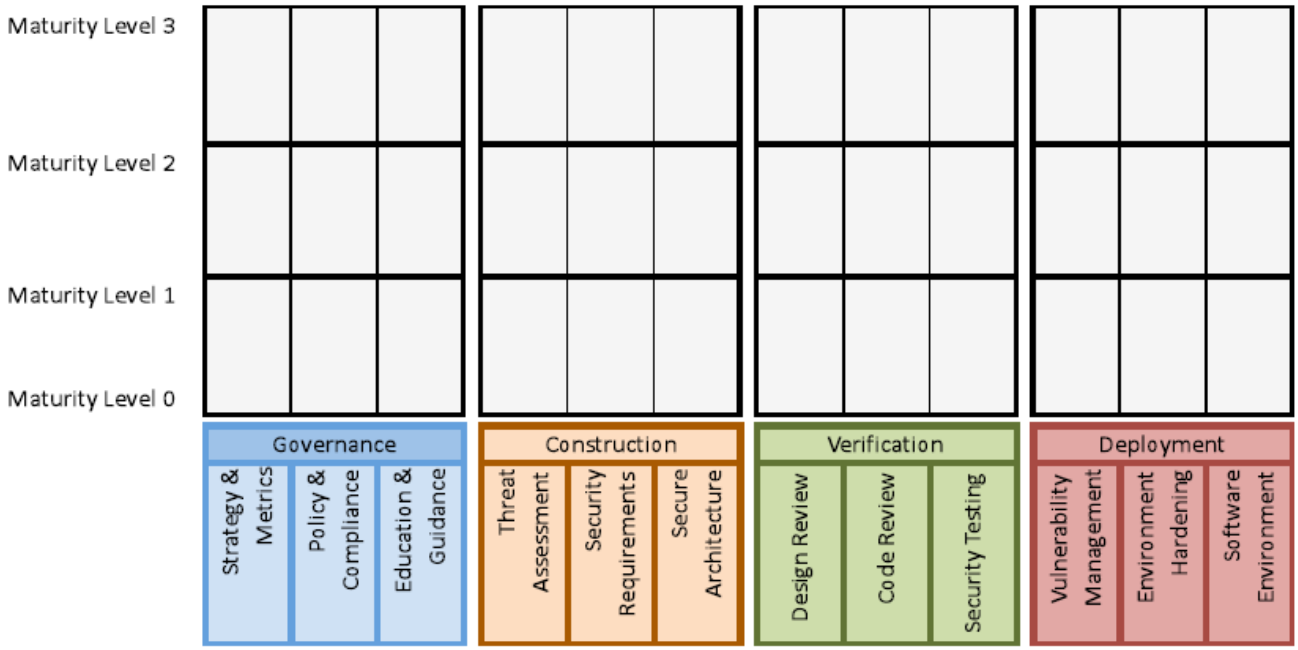


*Abbildung-5: Template für die Bewertung der vorhandenen bzw. zu erreichenden Reifegrade pro Security Practice*

wahrscheinlich nicht zu Unrecht, denn eine Reihe von Indikatoren bestätigt die Methode von Microsoft doch als effektive Vorgehensweise.

**Kontinuierliche Prozessverbesserung (Qualität, Reife) vs. Abarbeiten von Checklisten?**

Viele Publikationen konzentrieren sich auf tatsächlich festgestellte Sicherheitsprobleme mit Webanwendungen – sei es in Folge realer Angriffe auf ein produktives System oder als Ergebnis von Audits bzw. konkreten Vulnerability-Scans. Ergebnis sind zumeist Aufzählungen der häufigsten Sicherheitsprobleme in vorhandenen Webanwendungen und eine Klassifikation von Typen und Herkunft bzw. Ursachen im untersuchten Programm. Klassische Vertreter dieser Aufzählungen sind:

- Top 10 Bedrohungen von Webanwendungen (OWASP),
- Top 25 Most Dangerous Software Errors (CWE/SANS) oder
- 19 Deadly Sins of Software Security (Michael Howard, David Leblanc, John Viega).

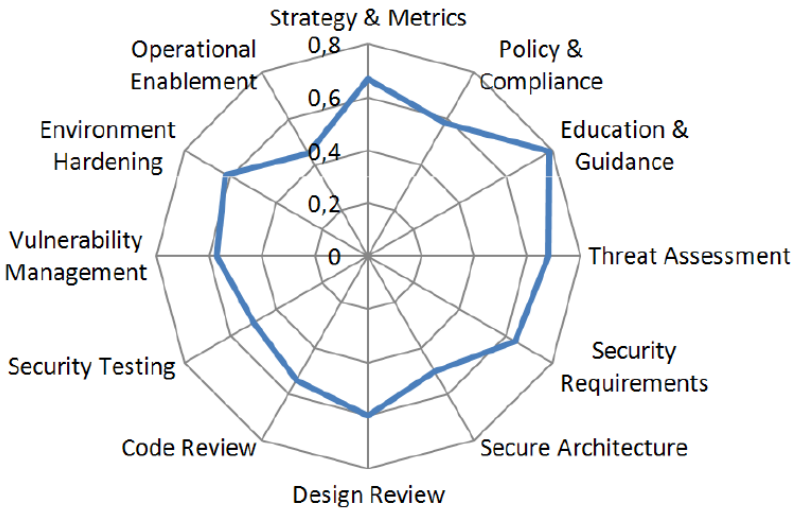


*Abbildung-6: Beispielhafte Darstellung der Ergebnisse einer konkreten Reifegradbeurteilung*

Selbst darüber hinaus gehende Best Practice-Sammlungen der OWASP oder z.B. der aus verschiedenen Herstellern gebildeten non-profit Organisation „Software Assurance Forum for Excellence in Code" (SAFECode), die sich um Vertrauensbildung für IuK Produkte und Dienstleistungen durch erhöhte Softwarequalität bemüht, können nur punktuelle Hinweise geben. Tatsächlich liegt es nahe, den Entwicklern genau die Hinweise auf konkrete Probleme zu geben, die entsprechend hoch priorisiert wurden und sie aufzufordern, geeignete Maßnahmen zur Vermeidung dieser Schwachstellen zu

ergreifen. Jedoch kann dies allein nicht zu einer systematischen Verbesserung der Qualität und Sicherheit führen, weil sämtliche Maßnahmen ausschließlich als Reaktion (=Beseitigung von Sicherheitsproblemen) auf erkannte oder sogar bereits wirksam gewordene Sicherheitsmängel erfolgen.

Wie ebenfalls aus Sicherheitsvorfällen erkennbar ist, greift die Gegenwehr bei Einzelmaßnahmen sehr oft auch viel zu spät – z.B. immer dann, wenn die Probleme durch unzureichende Konzepte, Qualitätssicherung / Testing u.a. bedingt sind. Hier muss Bewusstsein und Qualität in den Entwicklungsprozess schon ganz am Anfang implantiert werden, damit irgendwann einmal das Ziel erreichbar wird, dass Webanwendungen (Software allgemein) eine wie selbstverständlich erwartete – und tatsächlich gelieferte – Qualität und Sicherheit aufweisen, wie es bei anderen industriell gefertigten Produkten auch der Fall ist. Hier setzen z.B. die Reifegradmodelle an und erinnern an die Tugenden und Ansprüche des guten, alten Software-Engineerings – mit besonderem Fokus auf Sicherheit. Es hat im Alltag der Entwicklung von Webanwendungen gelegentlich den Anschein, als wäre Software-Engineering den Anforderungen des Marktes (time-to-market, Wettbewerb über Kosten etc.) geopfert worden und klassische Vorgehensmodelle würden lediglich als akademisches Ideal und nur für Großprojekte – womöglich wiederum im militärischen Umfeld – umsetzbar sein.
Es bleibt allerdings die begründete Hoffnung, dass mittelfristig die Kombination aus wachsendem (Sicherheits-)Bewusstsein, allgemeingültigen oder auch gesetzlich geregelten Standards und Vorgaben sowie die kommerzielle Notwendigkeit (unter Mitwirkung der Kunden und Nutzer) zu qualitativ höherwertiger und sicherer Software führen wird.

Wenn die Ausbildungsinstitute und Hochschulen ihren Beitrag dazu leisten, die Awareness für Sicherheitsthemen als festen Bestandteil der Softwareentwicklung zu verankern, könnte vielleicht bereits mit der kommenden Entwicklergeneration der „turnaround" im Wettrüsten zwischen Entwickler und Angreifer geschafft werden.

Mit dieser Begrifflichkeit und Wortwahl sind wir am Ende des Artikels – wie nicht selten bei der Behandlung von IT-Themen – wieder im militärischen Umfeld gelandet und bei der eingangs erwähnten Herkunft und Entstehung der Reifegradmodelle für den Softwareentwicklungsprozess. Dies soll keinesfalls als Aufforderung verstanden werden, den militärischen Bereich zum Vorbild zu nehmen – aber durchaus als Hinweis darauf, dass trotz vermeintlich unermesslicher Budgets dieses Bereichs nicht automatisch qualitativ bessere und sicherere Software entwickelt wird, es kommt auf die Qualität der Prozesse an.



**Referenzen**

[1] Introduction to Software Process Improvement, Watts S. Humphrey 1992
[2] SEI: Comparison of ISO 9001 and the Capability Maturity Model for Software
[3] openSAMM
[4] Top 25 Most Dangerous Software Errors (CWE/SANS)
[5] "19 Deadly Sins of Software Security" (Amazon oder eBook)
[6] Software Assurance Forum for Excellence in Code" (SAFECode)