Measuring How Grown-Up Your Process Is: CMMI, IDEAL, and SAMM

Fri, Sep 18, 6:00 PM EDT · https://scottslab.io/posts/software-maturity-models-cmmi-samm

TL;DR

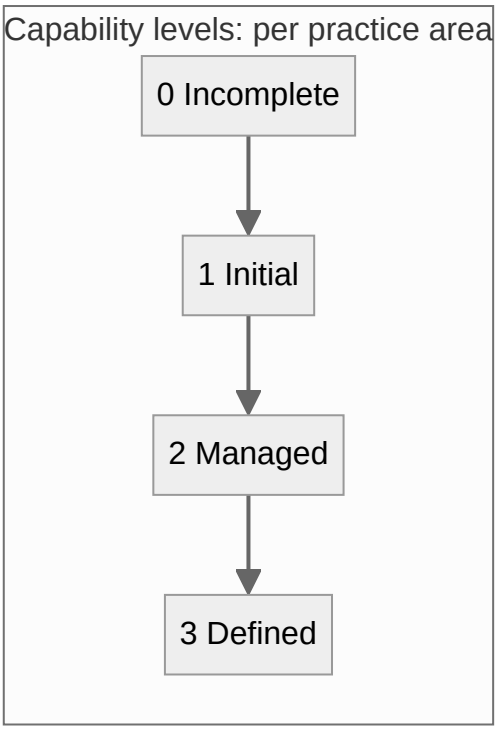
CMMI grades your process on two separate axes people usually collapse into one: six staged maturity levels for the organization (Incomplete through Optimizing, including a Level 0 most write-ups skip) and four capability levels per individual practice area. OWASP SAMM does the same thing for security specifically: five business functions, fifteen security practices, each scored on its own three-level maturity scale. Neither number is your actual security posture. It's a description of how institutionalized your practices are, and the failure mode every mature-sounding organization eventually hits is optimizing the score instead of the risk it's supposed to represent.

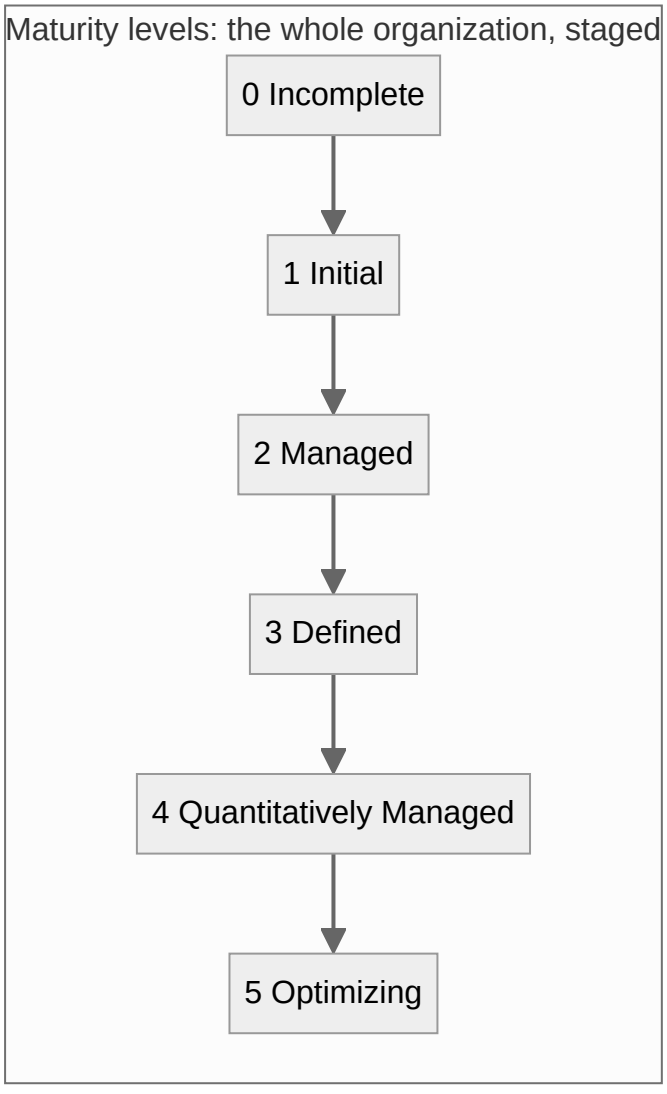
A methodology tells you how you build software. A maturity model tells you how reliably you do it: whether last quarter's discipline survives someone quitting, a deadline, or a reorg. Most summaries flatten these models into a single ladder; both of the serious ones are more precise than that, and the precision is where the useful part lives.

CMMI is two scales wearing one name

CMMI (Capability Maturity Model Integration, maintained by the CMMI Institute, now part of ISACA) is usually described as five levels. The current model actually defines six **maturity levels**, because Level 0 exists and gets left out of almost every summary: Incomplete, ad hoc and unknown, work may or may not get completed at all. Level 1, Initial, is unpredictable and reactive: work gets done, but late and over budget. Level 2, Managed, means projects are planned, performed, measured, and controlled, but only at the individual project level, so discipline doesn't yet survive a team boundary. Level 3, Defined, is where org-wide standards apply across projects, programs, and portfolios, so the practice doesn't depend on which team you're on. Level 4, Quantitatively Managed, means the organization is data-driven, with quantitative objectives that are actually predictable. Level 5, Optimizing, is stable enough to be flexible: continuous improvement built into operations, not a special initiative bolted on top.

Maturity levels describe the organization as a whole, in a fixed sequence: you can't hold a Level 4 without having genuinely built Level 3 underneath it. Alongside that, CMMI separately defines **capability levels**, which apply within a single practice area: Level 0 Incomplete, Level 1 Initial, Level 2 Managed, Level 3 Defined. This is the axis most casual descriptions miss, and it's the more useful one day to day. An organization can be genuinely strong in incident response and genuinely weak in requirements management at once, and a single maturity number would hide exactly that.


Capability levels: per practice area
0 Incomplete
1 Initial
2 Managed
3 Defined


Maturity levels: the whole organization, staged
0 Incomplete
1 Initial
2 Managed
3 Defined
4 Quantitatively Managed
5 Optimizing

CMMI is two scales wearing one name

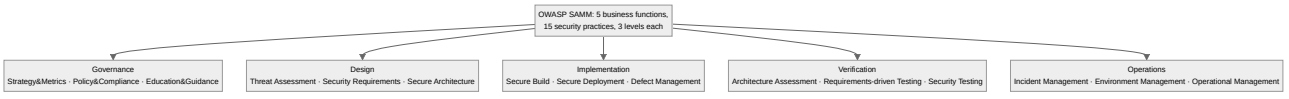
The appraisal method that actually produces a CMMI rating is SCAMPI (Standard CMMI Appraisal Method for Process Improvement), run by trained appraisers against defined practice areas, not a self-assessed checklist. That matters for the failure mode below: a real appraisal is expensive and infrequent enough that organizations have a strong incentive to prepare *for the appraisal* specifically.

IDEAL is the how, CMMI is the where

CMMI tells you where you are. It doesn't tell you how to move. That's what the SEI's IDEAL model is for: a five-phase improvement loop (Initiating, Diagnosing, Establishing, Acting, Learning) meant to structure an improvement effort, not just measure its outcome. Initiating sets the business case and gets sponsorship. Diagnosing assesses where you currently stand, often via a CMMI-style appraisal. Establishing turns the diagnosis into a concrete plan and priorities. Acting is where the process changes get made and piloted. Learning closes the loop: did it work, and what does the next cycle do differently. The two pair naturally: IDEAL is the verb, CMMI's levels are the destination.

SAMM does the same thing, scoped to security

OWASP's Software Assurance Maturity Model applies the identical logic: stop asking the unanswerable "are we secure?" and start measuring specific, gradeable practices. It just scopes that logic to security instead of process in general. SAMM 2.0 defines five business functions: Governance, Design, Implementation, Verification, and Operations. Each function contains exactly three security practices, for fifteen total. Governance covers Strategy & Metrics, Policy & Compliance, and Education & Guidance, for instance, while Design covers Threat Assessment, Security Requirements, and Secure Architecture. Every security practice is further split into two streams covering different aspects of that practice, and each stream is scored on its own three-level maturity scale, with the activities at each level more sophisticated and more strictly measured than the one below it.



SAMM does the same thing, scoped to security

That granularity is the whole point. SAMM deliberately doesn't insist every organization max out every practice. A startup and a bank should have different target profiles, and the model is explicit that you pick a target maturity per practice rather than chase a single overall number.

The number is not the thing it measures

Here's where every maturity program eventually goes wrong. A maturity level, at either model, describes whether a practice is *institutionalized*: repeatable, documented, measured, independent of any one person's memory. It says nothing directly about whether an attacker can get in. An organization can hit SAMM Level 3 in Security Testing, meaning testing is deeply embedded, automated, and metric-driven, while still shipping a critical vulnerability, because the tests that are rigorously run were never the ones that would have caught that particular bug. The score measures the process running the tests, not the coverage of what the tests actually check.

The predictable failure mode is optimizing the appraisal instead of the risk: writing the policy the assessor wants to see, scheduling the SCAMPI appraisal around a period when things look calm, building metrics dashboards that make Level 4 defensible without changing what the metrics are supposed to be driving. None of that is fraud, exactly. It's just what happens when the score becomes the target instead of the proxy it was designed to be. The honest use of either model is the reverse: let the gaps the assessment surfaces tell you where to actually spend, and treat the level itself as a status report to leadership, not a finish line.

Written by Scott S. — Contact: scott@scottschlangen.com — scottslab.io — https://scottslab.io/posts/software-maturity-models-cmmi-samm