Poprawa procesów programowych w małej organizacji programistycznej

Grzegorz Wieczerzak
Politechnika Poznańska

Streszczenie

Program poprawy procesów programowych powinien być dobrze zaplanowany. O sukcesie przedsięwzięcia najbardziej decyduje poparcie kadry kierowniczej firmy programistycznej, bowiem nie wszyscy inwestują w badania i długoterminowe działania. Doświadczenie jednak wskazuje, że przezwyciężenie strachu przed zmianami starych nawyków przynosi miarodajne korzyści. W dzisiejszym świecie, któremu towarzyszą szybkie zmiany ekonomiczne, socjologiczne i technologiczne, firmy wytwarzające oprogramowanie borykają się ze złą organizacją pracy, skarżą się na komunikację z klientem. W warunkach stresu, braku czasu i niewłaściwej interpretacji wymagań gasi się pożary. Z drugiej strony daje się zauważyć zaangażowanie pracowników w eksperyment, oraz potrzebę jasnego wizerunku firmy. Dzięki właściwej motywacji w krótkim czasie polityka zapobiegawcza firmy informatycznej ułatwi eksplorację nowych trendów technologicznych, a przez to pozyskanie nowych klientów.

1. Synergia trójwarstwowego modelu

„All models are wrong; some models are useful."

George Box

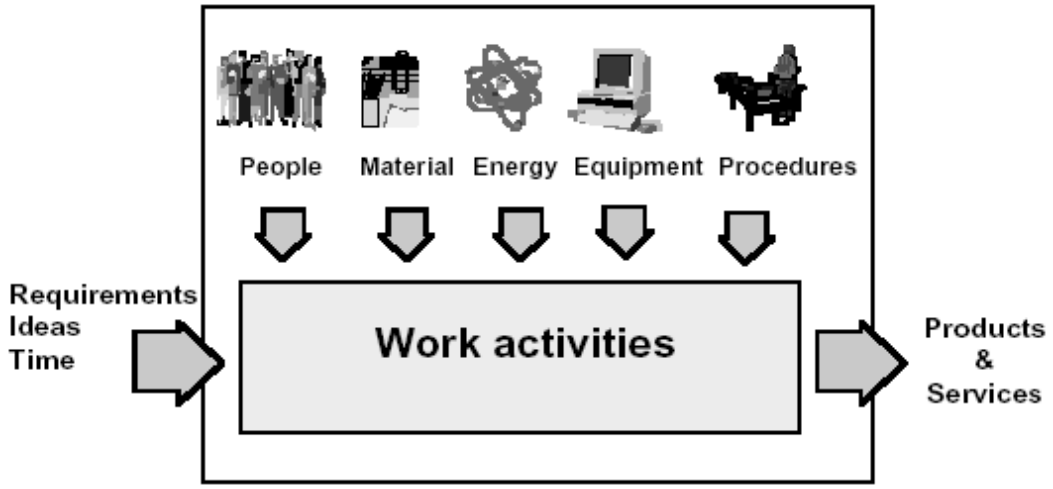
Rozwój firmy informatycznej jest w dzisiejszym świecie nieunikniony. Coraz częściej obserwuje się potrzebę podkreślenia jakości swoich produktów. Brak profesjonalizmu w działaniu może być nieprzewidywalny, łącznie z utratą klientów.

To klienci decydują o kształcie rynku. Dom, który został zbudowany w trzy tygodnie, może budzić obawy w potencjalnym lokatorze. Dlatego budownictwo objęte jest rygorystycznymi przepisami i normami. Według określonych standardów architekt zaczyna od szkicu obiektu, który następnie stopniowo uszczegółowia, mając na względzie życzenia i wymagania klienta, znając wymiary działki oraz zasobność portfela nabywcy.

Gotowy plan jest końcowym produktem procesu projektowania. Z tym planem możemy zrobić wiele rzeczy. Udać się z nim do firmy budowlanej, lub schować go do szuflady dla swoich dzieci. Ponadto, jeśli pomysł zbudowania domu spodoba się naszym znajomym, cały proces powtórzy się według podobnych zasad.

Wtargnęliśmy w tym momencie na zagadnienia związane z popularnym w ostatnich latach zarządzaniu relacjami z klientem, CRM (ang. *Customer Relationship Management*) i chociaż to nie o tym jest poniższe opracowanie, to jednak nacisk na stosunki międzyludzkie oraz na wartości powinien być wkalkulowany w każdą inicjatywę związaną z poprawą procesów. Przedsięwzięcie nie zakończy się sukcesem jeśli wyprodukujemy tonę procedur i standardów. Pora więc poznać właściwą interpretację pojęcia „proces".

Co to jest proces? Proces jest to sekwencja kroków wykonywanych w określonym celu. Proces integruje ze sobą ludzi, narzędzia i procedury.



Rys. 1 Definicja procesu.

Korzystając z odpowiedniego wyposażenia, narzędzi, metod postępowania zespół pracowników o różnym poziomie doświadczenia, przeszkolenia i motywacji jest w stanie przetwarzać surowe materiały w złożone produkty,

pełnowartościowe dla klienta. Warunek jest jeden. Opisane procesy należy wdrażać w życie.

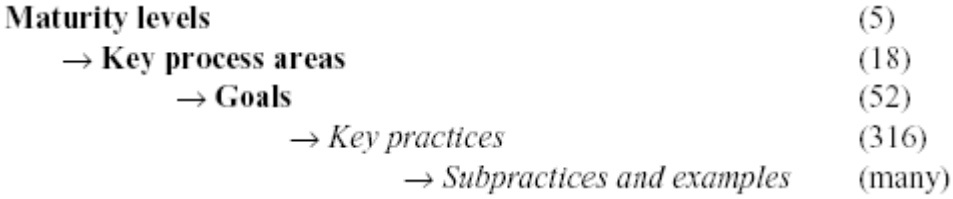
Problemem jest takie opisanie procesów, by były akceptowalne przez czynnik ludzki. Procedury i pewne czynności mają usprawnić naszą pracę, wnieść pewien porządek i w efekcie przyczynić się do dobrowolnego przyjęcia „dojrzałych" nawyków.

Skoro poznaliśmy trójwarstwową interpretację procesu, możemy spojrzeć z perspektywy firmy programistycznej. Procesy programowe, czyli związane z produkcją oprogramowania mogą być zdefiniowane jako zestaw czynności, metod, praktyk i transformacji, które są stosowane przez określone zespoły lub jednostki, w celu wytworzenia i pielęgnowania oprogramowania i produktów pośrednich, takich jak plany projektu, dokumentacja architektury systemu i wymagań, kod źródłowy, przypadki testowe, dokumentacja użytkownika, etc.

Pierwszym pojawiającym się pytaniem jest „od czego zacząć". Wiele organizacji, które podjęły trud osiągnięcia wyższego poziomu dojrzałości oprogramowania, swój sukces zawdzięcza zdefiniowaniu właściwego modelu poprawy procesów programowych. Najbardziej znanym wzorcem dojrzałości oprogramowania jest model CMM[SM], SW-CMM (ang. *Capability Maturity Model for Software*) zdefiniowany w Instytucie Inżynierii Oprogramowania, na Uniwersytecie Carnegie Mellon w Pittsburgu.

2. Skuteczny program poprawy

Prawidłowe używanie modelu SW-CMM oznacza równoważenie przeciwstawnych celów. Szacowanie poziomu dojrzałości organizacji wymaga fachowej oceny sytuacji. Model SW-CMM dostarcza znacznej ilości porad przy dokonywaniu takich osądów. Hierarchiczna struktura CMM, ułożona w kluczowe obszary procesów, które definiują cele, zwyczaje (ang. *key practices*) i praktyki podrzędne (ang. *subpractices*), jest łatwa w interpretacji i pozwala na dostosowanie modelu dojrzałości do potrzeb organizacji.



Rys. 2 Hierarchiczna interpretacja modelu CMM.

Model SW-CMM definiuje pięć poziomów które są wyznacznikiem dojrzałości danej organizacji programistycznej. Poczynając od poziomu pierwszego, początkowego, który charakteryzuje się brakiem procesów i chaotycznym działaniem przy poszczególnych etapach przedsięwzięcia, przechodzimy na poziom powtarzalny, ukierunkowany na procesy związane z zarządzaniem projektami. Na poziomie drugim poszczególne czynności wykonywane są w sposób podobny. Pojawia się w związku z tym szansa na powtórzenie sukcesu przy realizacji projektu o podobnym charakterze.

Rozpoczynając program poprawy procesów programowych, w pierwszej kolejności organizacja powinna dążyć do opanowania następujących aspektów zarządzania projektem:

- Zarządzanie wymaganiami
- Planowanie projektów programistycznych
- Śledzenie realizacji projektów
- Zarządzanie podzleceniami
- Zapewnienie jakości oprogramowania
- Zarządzanie konfiguracją oprogramowania

Poprawa powinna objąć jednocześnie jak najwięcej obszarów, zaczynając od najprostszej definicji procesów. Pozwoli to na szybkie przystosowanie sprawdzonych i pożytecznych procesów do zwykłych, codziennych projektów realizowanych w organizacji.

Projekt polega na poprawie nieefektywnych nawyków. Dlatego należy wypunktować wszystkie przyjęte procesy w takiej postaci, w jakiej są w rzeczywistości, a nie jakiej powinny być. Model CMM pod tym względem okazuje się bardzo pomocny, ponieważ nie jest dokładną, uniwersalną instrukcją. CMM jest pożytecznym narzędziem w prowadzeniu projektu SPI. Nadaje się on pod tym względem do zastosowania w dużych organizacjach realizujących projekty rządowe, jak też w małych organizacjach, produkujących nieduże aplikacje.

2.1. Wsparcie kadry kierowniczej

Sponsorowanie jest nieodłącznym elementem badań. Gdy mówimy o poprawie procesów w firmie, wsparcie ekonomiczne, a także moralne jest kluczowym elementem w projekcie budowania dojrzałości organizacji. Jako jednostki samodzielne, pracownicy dzięki nabytym doświadczeniom, objawiają profesjonalizm w danej dziedzinie. Potrafią też niekiedy, w ramach własnej strefy wpływów, zarządzać dyscypliną pracy. Inaczej jest, gdy spojrzymy na organizację jako całość, która dąży do polepszenia wyników działalności. Wówczas zmiany

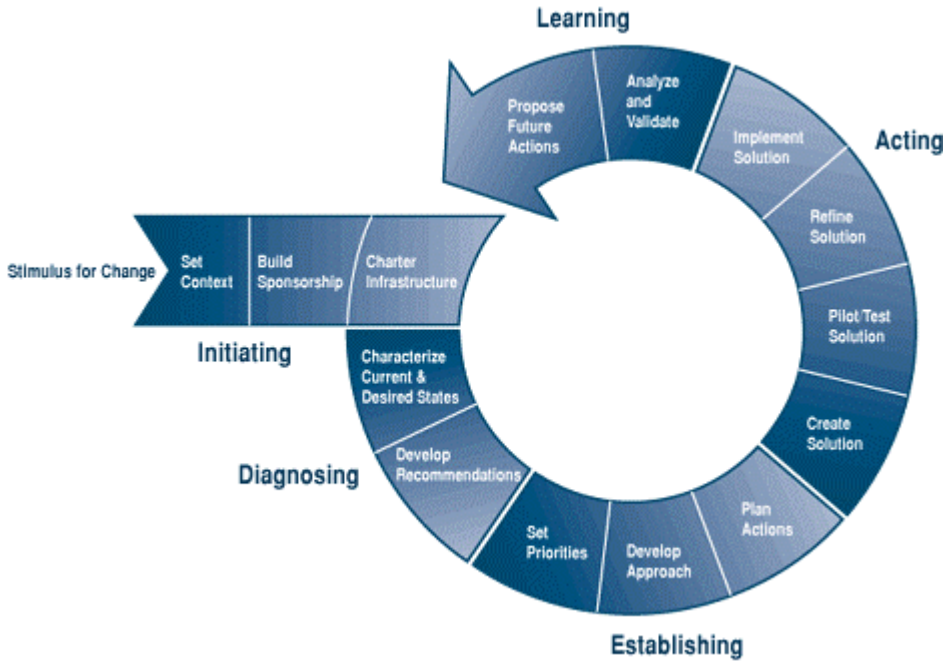
muszą być aktywnie wspierane przez jej kierownictwo. Projekt poprawy procesów bez niezbędnej pomocy i koordynacji od góry, nie przyczyni się do zbudowania dojrzałości w całej organizacji. Zamiast niej pojawią się zamknięte obszary, w których firma się wyspecjalizuje, tzw. wyspy doskonałości.

Wsparcie kierownictwa polega na zdefiniowaniu misji, polityki działania oraz dostarczaniu zasobów niezbędnych do realizacji procesów programowych. Nieocenione także jest przekonanie i wiara w strategiczne znaczenie poprawy procesów. Wskazane jest aby szczeble kierownicze firmy okazywały poparcie dla działań zespołów zaangażowanych w projekt oraz zwracały uwagę i wyrażały zainteresowanie zachodzącymi przemianami.

W liczących się organizacjach, na Zachodzie Europy i w USA, dodatkowym czynnikiem motywującym wszystkich pracowników, jest podkreślanie przez zarząd znaczenia jakości dla firmy i jej klientów. Okazją do tego są wspólne spotkania świąteczne, noworoczne, wewnętrzne konferencje w obecności przedstawicieli klientów, itp. Dzięki otwartej postawie, kierownictwo zyskuje respekt w oczach pracowników, a wszelkie kryzysy, w rodzaju: skrócenie terminu realizacji projektu, zmniejszenie budżetu, pogorszenie warunków pracy, które mają prawo zdarzyć się w każdej organizacji z różnych przyczyn, nie będą odbierane wyłącznie jako problemy zarządu.

Wspieranie projektu poprawy w perspektywie długoterminowej powinno iść w parze z celami ekonomicznymi organizacji. Program doskonalenia zdolności procesów powinien mieć wysoki priorytet. Dzięki temu kadra kierownicza będzie w stanie monitorować ciągłe, systematyczne zmiany. Zdobywanie wyższych poziomów dojrzałości jest działaniem przyrostowym i wymaga długoterminowego zaangażowania. Ważna jest umiejętność stawiania jasnych celów i prowadzenia proaktywnej polityki. Do poprawy procesów powinno podchodzić się jak do normalnego projektu. Przygotować plan, scharakteryzować kamienie milowe, wyznaczyć miary postępu.

Poprawa oparta na modelu CMM jest najefektywniejszym rozwiązaniem, choć sam model nie powinien być traktowany jako metodologia postępowania czy planowania projektu SPI. Instytut SEI (CMU) proponuje w tym celu systematyczne podejście do projektu poprawy, jakim jest IDEAL^{SM}. Model IDEAL (patrz. rys. 3) jest pięciofazowym cyklem wyznaczającym ścieżkę realizacji projektu SPI. Możemy go również interpretować jako model składający się z dwóch etapów: strategicznego, bazującego na potrzebach organizacji oraz taktycznego, w którym procesy są definiowane, modyfikowane i uruchamiane. Po zakończeniu obu faz, następuje etap pozyskiwania wiedzy *Learning*, podczas którego kierownictwo definiuje nowe cele i strategie.



Rys. 3 Cykl wdrażania projektu poprawy procesów, IDEAL.

2.2. Zarządzanie przez fakty

Wszelka analiza, którą przeprowadzamy na danych, powinna być użyteczna. Aby osiągnąć zamierzony rezultat, należy zrozumieć, jakie informacje są nam przekazywane i w jaki sposób można je interpretować. Proces definiowania miar rozpoczyna się od zebrania prostego zbioru użytecznych danych. Następnie wprowadza się standardowe definicje dla typowych miar, co pozwala na agregowanie i porównywanie ich wartości z różnych środowisk.

Następstwem zbierania danych jest wiara w ich znaczenie. Jeśli na przykład z analizy wynika, że programiści są w stanie osiągnąć wydajność 100 LOC/miesiąc, nie uda się, bez zaangażowania dodatkowych środków, spełnić wymagania klienta, który narzuca nam tempo pracy 1000 LOC/miesiąc. Należy wówczas zająć odpowiednie stanowisko w uzgodnieniu z klientem i członkami zespołu. Zmienianie zobowiązań i funkcjonalności bez tego porozumienia nie służy poprawie procesów.

Do interpretacji miar należy podchodzić z dużą czujnością. Zła interpretacja danych, może przyczynić się do obrania niewłaściwej drogi poprawy. Kierownictwo wyznaczając pewne cele i kierunki, tym samym przekazuje pracownikom do wiadomości, co jest w danej chwili najważniejsze. Właściwe

zarządzanie i kierowanie dowolnym przedsięwzięciem, powinno opierać się na miarach. Dzięki temu można uniknąć sytuacji, w których skupiamy cały wysiłek na harmonogramowaniu, spychając jakość na drugi plan. Miary określają, co jest ważne. Oczywiście nie wszystko można w prosty sposób śledzić. I właśnie na te elementy trzeba zwracać szczególną uwagę.

Organizacja powinna stosować politykę zobowiązań opartą na zasadzie „robimy, co umiemy". Pracownicy angażują się osobiście przy gaszeniu nagłych „pożarów", jeśli kierownictwo podejmuje realistyczne zobowiązania wobec klienta, z uwzględnieniem możliwości produkcyjnych zespołu.

2.3. Wybór istotnych procesów

Wiele organizacji prowadzących projekt poprawy procesów, wydzieliło spośród pracowników zespół odpowiedzialny za zarządzanie procesami programowymi, SEPG (ang. *Software Engineering Process Group*). Utworzenie grupy SEPG, lub równoważnej, powinno nastąpić dosyć wcześnie, np. w fazie początkowej cyklu IDEAL. Na jej członków należy wybrać osoby kompetentne oraz cieszące się autorytetem, pełniące kierownicze obowiązki, a także pracownicy „techniczni". Bardzo ważne jest, aby osoby te posiadały dobre umiejętności interpersonalne. Sukces zespołu SEPG zależy od zdolności komunikacji, uczenia innych, negocjowania i konsultacji.

Grupa do spraw poprawy procesów powinna mieć, w ramach organizacji, dużą swobodę w dysponowaniu zasobami. Z drugiej strony, na zespole SEPG spoczywa obowiązek wykonywania akcji wynikających z oceny bieżącej sytuacji. Wiele programów poprawy zakończyło się tylko na dobrych intencjach. Poprawa jest następstwem planowania czynności, przydzielania zasobów do zadań i ciągłej kontroli programu.

Ponieważ działania osób zaangażowanych w kontrolę procesów pochłaniają dużo wysiłku i czasu, zespół SEPG powinien być otwarty dla wszystkich. Co pewien czas wskazane jest uczestnictwo w projekcie nowych pracowników. Na czym polega koordynowanie prac przez zespół ds. procesów? Dzięki angażowaniu w program dobrze przygotowanych pracowników, ich różnorodna wiedza i doświadczenie umożliwia rozwiązywanie trudnych problemów, przed którymi stoi organizacja. Jeśli w firmie proces poprawy procesów i jakości będzie wynikał ze wspólnej pracy, zapewniony będzie sukces ciągłego rozwoju, zarówno oprogramowania jak i dojrzałości na poziomie procesów organizacji, w rozumieniu modelu TQM (Total Quality Management).

Definiując procesy należy mieć na uwadze dyscyplinę, jaką narzucają. Nie można też zapominać o kreatywności pracowników, którzy nie zawsze chcą podporządkować się sztywnym regułom. To samo dotyczy niektórych projektów,

które czasem będą wykraczały poza standardowy zestaw procedur. Pozostawienie pewnego marginesu swobody zapobiega zniechęceniu programistów do stosowania się do procesów.

O użyteczności i efektywności procesów decydują następujące atrybuty:

- Powinny być **udokumentowane**. Przy opisie należy używać prostego, jednoznacznego języka.
- Każdy pracownik powinien być **przeszkolony** w zakresie danego procesu.
- Są **stosowane** podczas codziennych czynności związanych z produkcją oprogramowania. Nie należy ich porzucać w chwilach kryzysu.
- Powinny być **mierzone** za pomocą prostych miar, które dostarczają odpowiedzi na krytyczne pytania.
- Wyznaczone osoby są odpowiedzialne za **utrzymywanie** i **wspieranie** procesów.
- Zmiany procesu są **kontrolowane**.
- Są **weryfikowane** i **akceptowane**.

Procesy należy definiować prostym i zwięzłym językiem. Ich opis nie musi być długi i złożony. Model SW-CMM wspomaga wykonywanie czynności, a nie posiadanie przepisów. Opis 1-2 stronicowy procesu może w zupełności wystarczyć. Z kolei podprocesy i podprocedury są wywoływane, gdy zajdzie taka potrzeba. Celem programu poprawy procesów nie jest zbudowanie złożonego mechanizmu sterowania. Porządek organizacji tworzy zbiór kilku kluczowych formuł i zasad.

2.4. Od ryzyka do jakości

Zarządzanie projektami programistycznymi jest ściśle związane z zarządzaniem ryzykiem. SEI definiuje ryzyko jako możliwość poniesienia straty. Właściwe podejście do czynników ryzyka może zmniejszyć skalę popełnianych błędów. Już w fazie zbierania wymagań, dąży się do unikania nieścisłości, co pozwala na łatwiejsze planowanie. Zdarza się, że oczekiwania klienta zmieniają się bardzo często. Firma programistyczna ustala więc procesy, które wspomagają zarządzanie ryzykiem dynamicznego otoczenia. Same procesy są jednak nie wystarczające. Organizacja powinna śledzić zmiany dyktowane przez świat.

Pojawia się często potrzeba zdefiniowania cyklu życia w organizacji. Pytanie o to, który model wytwarzania oprogramowania jest najlepszy, jest nieprecyzyjne. Przy realizacji projektów wymagających nowatorskich rozwiązań należy używać przyrostowy lub ewolucyjny cykl życia. Jeśli chcemy skierować wysiłek na zarządzanie ryzykiem, preferowany staje się model spiralny.

Rys. 4 Zarządzanie ryzykiem wplecione w cykl życia oprogramowania.

Gdy istnieje potrzeba częstych kontaktów z klientem, lub gdy odbiorca wymaga szybkich efektów pracy, warto pokusić się na stworzenie spójnego prototypu systemu. Długoterminowe projekty można łatwo zaplanować stosując model kaskadowy – prawdopodobnie nadal najpowszechniej stosowany. Każdy etap ma zdefiniowane produkty wyjściowe, łatwe jest również zarządzanie zasobami. Model kaskadowy znajduje również zastosowanie dla małych, kilkudniowych projektów.

3. Co oznacza ciągły rozwój

Pozyskiwanie jakości jest procesem długotrwałym. Organizacja nie poprzestaje na kolejnym poziomie CMM. Należy pamiętać dlaczego poprawa procesów jest ważna dla polityki firmy. Standardy i modele takie jak SW-CMM są pomocne w usprawnieniu procesów wytwarzania oprogramowania, jednak skupienie uwagi wyłącznie na zdobyciu wyższego poziomu dojrzałości bez faktycznej poprawy istniejących nawyków kończy się fiaskiem.

Poziomy CMM są miarą poprawy a nie celem samym w sobie. Pomyślne wdrożenie norm jakości i procesów programowych jest wynikiem wielostronnego zaangażowania.

Bibliografia

[1]. M.C. Paulk. *Using the Software CMM® in Small Organizations*. Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, 1998.

[2]. M.C. Paulk. *Effective CMM-Based Process Improvement*. Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, 1996.

[3]. B. McFeeley. *IDEAL^{SM}: A User's Guide for Software Process Improvement,* (CMU/SEI-96-HB-001). Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, February 1996.

[4]. M.C. Paulk, B. Curtis, M.B. Chrissis, C.V. Weber. *Capability Maturity Model^{SM} for Software, Version 1.1,* (CMU/SEI-93-TR-024). Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, February 1993.

[5]. M.C. Paulk, C.V. Weber, S.M. Garcia, M.B. Chrissis, M. Bush. *Key practices of the Capability Maturity Model^{SM}, Version 1.1,* (CMU/SEI-93-TR-025). Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, February 1993.

[6]. W.A. Florac, R.E. Park, A.D. Carleton. *Practical Software Measurment: Measuring for Process Management and Improvement,* (CMU/SEI-97-HB-003). Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, April 1997.

[7]. T.G. Olson, N.R. Reizer, J.W. Over. *A Software Process Framework for the SEI Capability Maturity Model^{sm},* (CMU/SEI-94-HB-01). Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, September 1994.