Pomiar jakości oprogramowania

ISO/IEC 9126 i SQueRE

ISO && IEC

Międzynarodowa Organizacja Normalizacyjna (ISO**)** tak jak **Międzynarodowa Komisja Elektrotechniczna** (IEC) są pozarządowymi organizacjami zrzeszającymi krajowe organizacje normalizacyjne.

Tworzą one wspólnie komitety techniczne współpracujące ze sobą w celu przygotowania norm międzynarodowych. Przygotowane projekty są przesyłane do komisji krajowych i tam poddawane głosowaniu. Aby normy zostały przyjęte muszą zaakceptować je przynajmniej 3/4 zrzeszonych krajów.

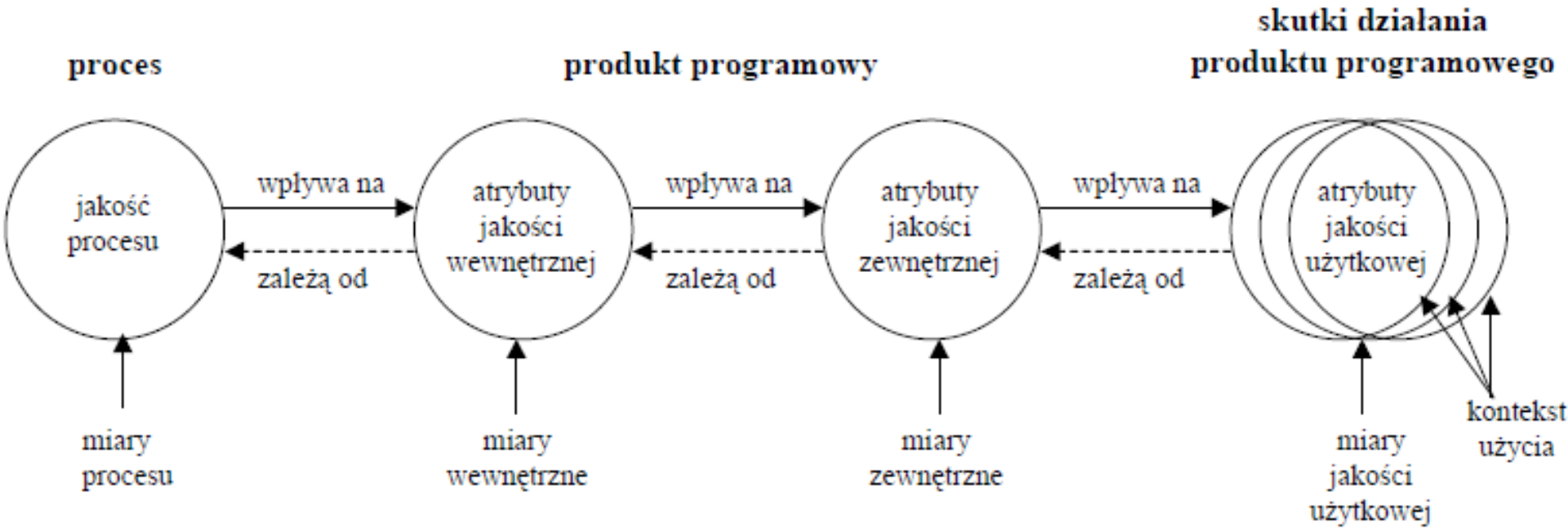
ISO/IEC-9126

9126 jest zbiorem 4 dokumentów wprowadzającymi standardy i sposoby oceniania procesu produkcji oprogramowania. W skład wchodzą następujące dokumenty:

- Część 1. Model jakości
- Część 2. Miary zewnętrzne
- Część 3. Miary wewnętrzne
- Część 4. Miary jakości użytkowej

Opisują one system oceny jakości oprogramowania w czasie jego tworzenia.

ISO/IEC-9126


proces
produkt programowy
skutki działania produktu programowego
jakość procesu
wpływa na
zależą od
atrybuty jakości wewnętrznej
wpływa na
zależą od
atrybuty jakości zewnętrznej
wpływa na
zależą od
atrybuty jakości użytkowej
miary procesu
miary wewnętrzne
miary zewnętrzne
miary jakości użytkowej
kontekst użycia

ISO 9126 & ISO 14598

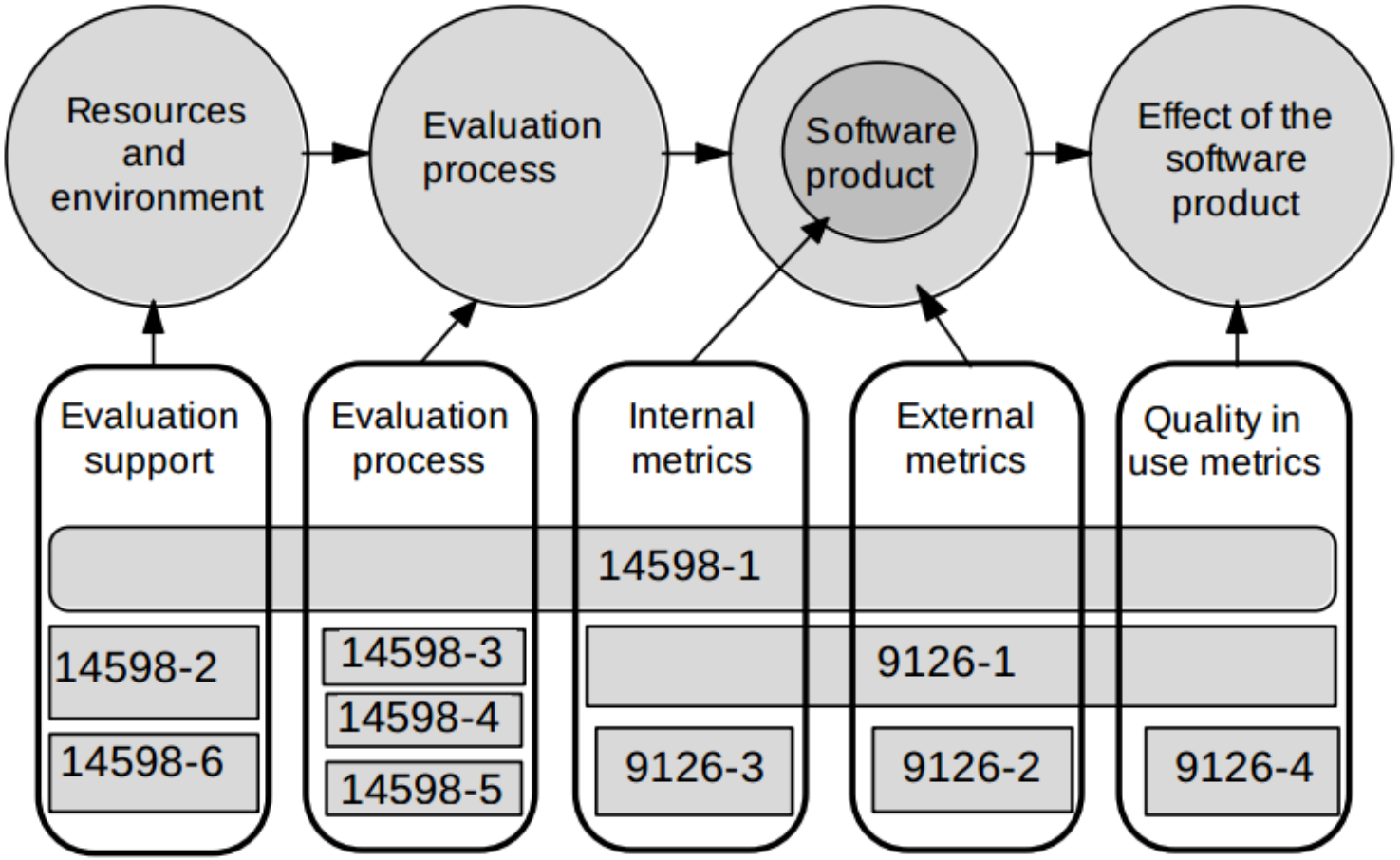


Figure 1: Current architecture of ISO/IEC 9126 and 14598 series

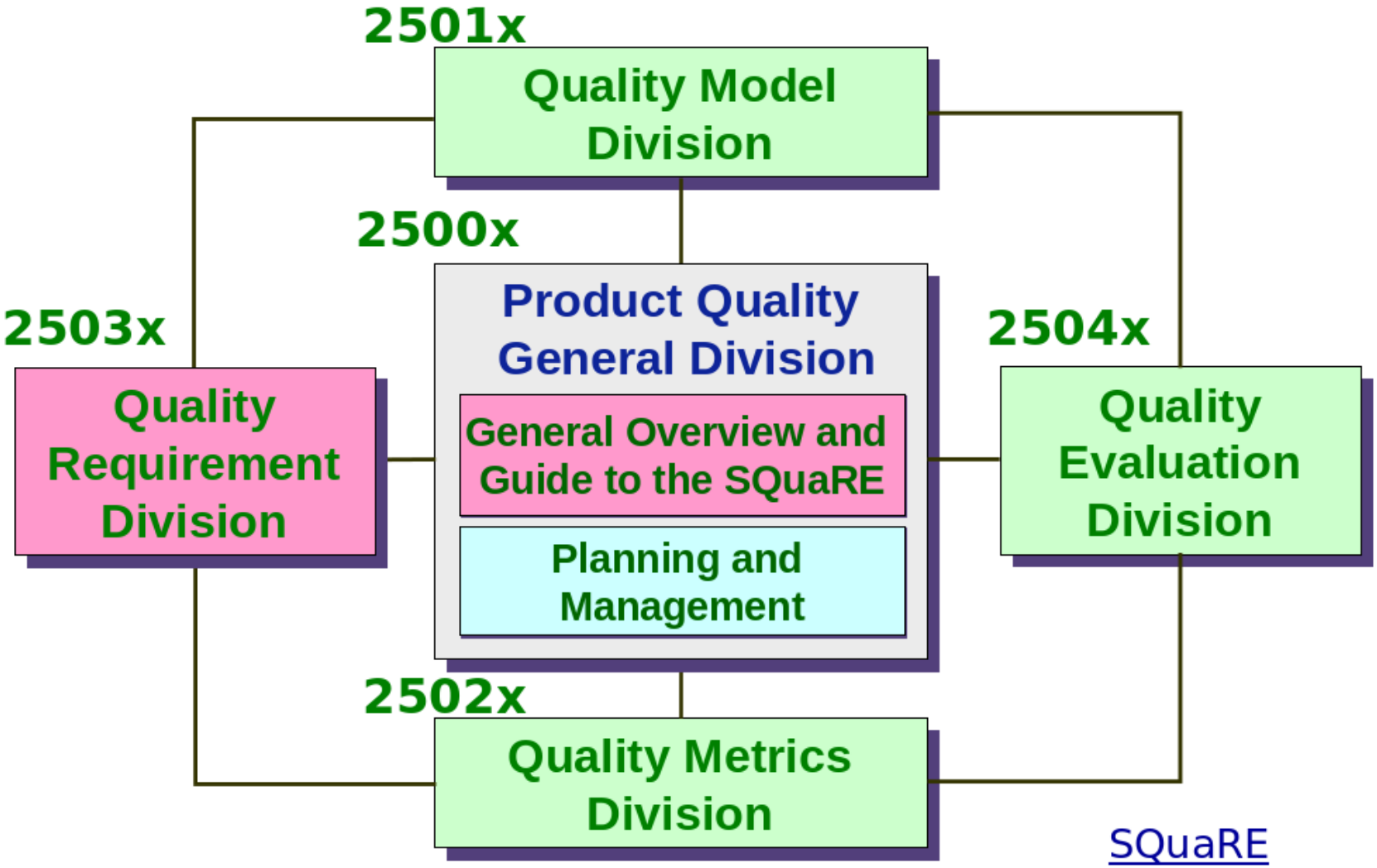
SQuaRE

Seria ISO/IEC 25000 (SQuaRE) została utworzona by zastąpić normę ISO/IEC 9126 oraz ISO/IEC 14598.

Składa się z następujących serii dokumentów :

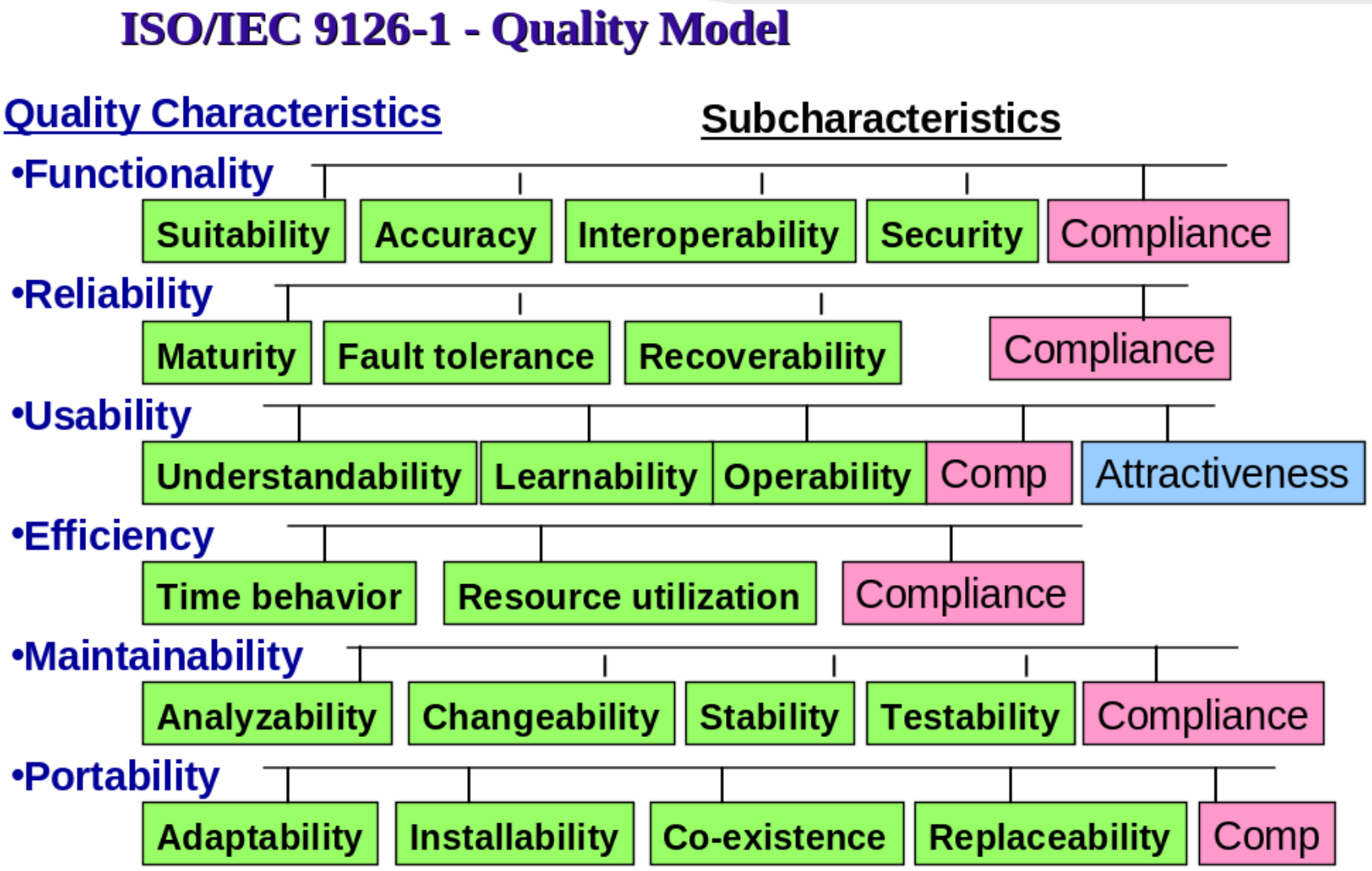
- Dział zarządzania jakością – 2500n
- Dział modelu jakości – 2501n
- Dział pomiarów jakości – 2502n
- Dział wymagań dla jakości – 2503n
- Dział oceny jakości – 2504n.

SQuaRE

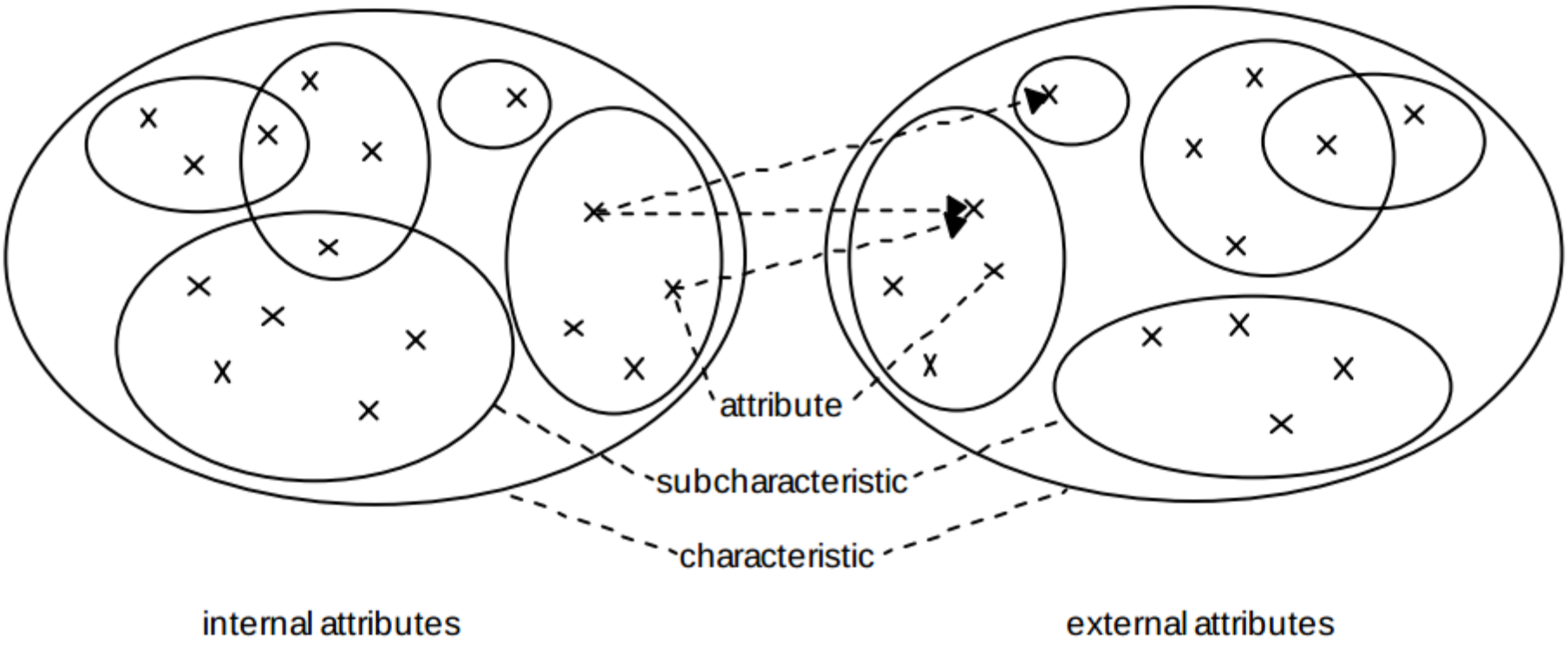


SQuaRE

Model jakości



Relacja między atrybutami



Funkcjonalność/odpowiedniość

Functional implementation completeness

Jak duża część specyfikacji została zaimplementowana

X = 1 - A/B

A -> liczba brakujących funkcji

B -> liczba już zaimplementowanych funkcji z specyfikacji

Funkcjonalność/odpowiedniość

Functional specification stability

Jak stabilna jest specyfikacja w cyklu rozwoju programu.

X = 1 - A/B
A -> liczba funkcji które wymagały zmian
B -> liczba funkcji opisanych w specyfikacji

Funkcjonalność/dokładność

Computational Accuracy

Jak często użytkownik napotkał niepoprawne wyniki.

X = A/T
A -> liczba błędnych wyników
T -> czas pracy aplikacji

Funkcjonalność/dokładność

Precision

Jak kompletnie wykonano wymagania precyzji.

X = A/B
A -> liczba funkcji spełniających kryteria
B -> liczba funkcji opisanych w specyfikacji

Funkcjonalność/współdziałanie

Data exchangeability(User's success attempt based)

Jak często komunikacja z innymi programami nie zawodzi.

Y= 1 – (A / B)
A -> liczba napotkanych niepowodzeń w komunikacji
B -> łączna liczba podjętych prób

Funkcjonalność/bezpieczenstwo

Data corruption Prevention

Jak często może wystąpić uszkodzenie danych programu.

X= 1 – A / N
Y= 1 – B / N
A -> liczba pomniejszych uszkodzeń danych
B -> liczba poważnych uszkodzeń danych
N -> łączna ilość testów

Poprzez pomniejsze uszkodzenie rozumiemy awarię po której dane można odtworzyć.

Funkcjonalność/bezpieczeństwo

Data encryption

Jak kompletna jest implementacja szyfrowania danych.

X=A/B
A -> ilość danych podlegających szyfrowaniu
B -> łączna ilość danych podlegających ochronie

Niezawodność/dojrzałość

Failure density (Fault density)

Natężenie występowania awarii (defektów).

X= NFAI / SIZE
Y= NFAU / SIZE
NFAI -> liczba wykrytych defektów
NFAU -> liczba wykrytych awarii
SIZE -> rozmiar projektu

Dzieląc wartości przez ilość testów możemy ocenić skuteczność testów

Niezawodność/dojrzałość

Mean time between failures (MTBF)

Średni czas między awariami.

X = TOPT / NAFI
Y = TSIB / NAFI
TOPT -> czas działania
TSIB -> suma interwałów miedzy awariami
NAFI -> łączna liczba zaobserwowanych awarii

Niezawodność/tolerancja

Breakdown avoidance

Jak często user może uniknąć 'zwisu' mimo awarii w oprogramowaniu

X= 1-(A / B)
A -> liczba 'zwisów'
B -> liczba awarii

Poprzez 'breakdown' rozumiemy sytuacje w której program wymaga zamknięcia lub restartu sprzętowego.

Niezawodność/wznawialność

Availability

Czy user może używać systemu przez dostatecznie duży czas ?

X= { To / (To + Tr) }
To -> czas działania
Tr ->czas napraw

Ta metryka powinna bazować na czasie automatycznej naprawy oprogramowania, nie zaś interwencji ludzkiej.

Użyteczność/łatwość-zrozumienia

Metryki bazujące na ocenie użytkownika

- Completeness of description
- Evident functions
- Function understandability
- Understandable Input and Output

Użyteczność/nauczalność

Metryki bazujące na ocenie czasu jaki poświęcił użytkownik

- Ease of function learning
- Ease of use of help system
- Tutorial Readiness
- Completeness of user documentation

Użyteczność/łatwość-obsługi

Metryki typu X=A/B :

- Input validity checking
- Customisability
- User operation cancellability
- Operation status monitoring capability
- Message Clarity
- Interface element clarity
- Self-explanatory error messages

Użyteczność/łatwość-obsługi

Time Between Human Error Operations

Czy user może używać systemu przez dostatecznie długo bez popełnienia błędów w obsłudze ?

X= T / N
T -> czas działania
N -> liczba błędów użytkownika

Poprzez błędy użytkownika rozumiemy:

- złe zrozumienie obsługi
- proste pomyłki
- dłuższe zastanowienie się nad kolejnym krokiem

Użyteczność/atrakcyjność

Attractive interface

Czy interface podoba się użytkownikowi

Brak wzorka :)

Użyteczność/atrakcyjność

User Interface appearance customisability

Możliwości użytkownika względem dostosowywania interface-u.

X=A/B

A-> liczba elementów
które można dostosować
B-> całkowita liczba
elementów w interface-ie

Efektywność/czas-działania

Throughput time

Ile zadań może być wykonane w danym przedziale czasowym ?

X = A / T

A -> liczba wykonanych zadań
T -> łączny czas testu

Efektywność/czas-działania

Mean response fulfillment ratio

Średni stosunek faktycznego czasu reakcji do czasu oczekiwanego

X = Tmean / TXmean
Tmean -> Σ(Ti) / N
TXmean-> oczekiwany średni czas
Ti -> czas i-tej próby
N -> łączna ilość prób

Efektywność/czas-działania

Worst case response time ratio

Czy w najgorszym przebiegu użytkownik uzyska odpowiedź w akceptowalnym czasie ?

X= Tmax / Rmax
Tmax -> **MAX(Ti)**
Rmax -> oczekiwane górne ograniczenie
MAX(Ti) -> maksymalny czas odpowiedzi w przeprowadzonych testach

Przy testach program powinien być poddany obciążeniu symulującemu docelowe warunki pracy.

Efektywność/użycie-zasobów

Metryki opisujące użycie zasobów mierzą ich widoczne wykorzystanie, średnie oraz górne ograniczenia. Dotyczą np. :

- **I/O**
- **Pamięci**
- **Przepustowości**

Utrzymanie/wykrywanie-błędów

Metryki sprawdzające poziom ułatwień przy debugowaniu pytają głównie o wykorzystanie :

- **Activity recording**
- **Readiness of diagnostic function**
- **Diagnostic function support**
- **Data recording during operation**

Utrzymanie/wykrywanie-błędów

Throughput time

Czy użytkownik może łatwo znaleźć przyczynę awarii ?

X= Sum(T) / N
T= Tout - Tin

Tout -> czas w którym odnaleziono przyczynę
Tin -> czas zgłoszenia awarii przez program
N -> ilośc awarii

Nie wliczamy awarii których przyczyn jeszcze nie odnaleziono.

Utrzymanie/łatwość-zmian

Time spent to implement the change for user's satisfaction

Ile czasu zajmuje wprowadzenie wymaganych przez usera zmian.

X = Sum(Tu) / N
Tu= Trc - Tsn
Tsn -> czas po którym user wysłał prośbę o zmiany
Trc -> czas do wydania kolejnej wersji programu
N -> ilośc wydanych wersji

Utrzymanie/stabilność

Less encountering failures after change

Czy użytkownik może używać programu dłużej bez awarii po aktualizacji ?

X= Na / Ta

Na -> liczba zaobserwowanych awarii
Ta -> czas testowania aplikacji po aktualizacji

Proponuje się porównywać ten współczynnik przed i po wprowadzeniu aktualizacji.

Utrzymanie/testowanie

Metryki wewnętrzne opisujące progres implementacji testów pytają o :

- Completeness of built-in test function
- Autonomy of testability
- Test progress observability

Utrzymanie/testowanie

Effortless testing

Czy użytkownik może łatwo przeprowadzić testy aby ocenić gotowość oprogramowania do pracy.

X= Sum(T) / N
N -> ilość usuniętych błędów
T -> czas poświęcony na usuwanie błędów

Znów pomijamy otwarte problemy.

Przenośność/adaptacja

Metryki wewnętrzne opisujące zdolność adaptacji programu :

- Porting user friendliness
- Hardware environmental adaptability
- Organisational environment adaptability
- Porting user friendliness
- System software environmental adaptability

Przenośność/adaptacja

User effortless adaptation

Czy użytkownik może łatwo dostosować program do środowiska.

X=T
T -> czas poświęcony na dostosowanie strony

Przenośność/instalacja

Metryki wewnętrzne opisujące instalajcę :

- Ease of Setup Re-try
- Installation effort
- Installation flexibility

Przenośność/instalacja

Installation easiness

Łatwość instalacji programu

X= A
A -> określane kategoriami :

- instalacja wymaga wykonania jednego programu (**znakomita**)
- dostarczona jest instrukcja instalacji (**dobra**)
- kod źródłowy musi być modyfikowany aby dostosować instalujące (**kiepska**)

Przenośność/koegzystencja

Concurrent multiple software use with less constraints

Jak często pojawiają się problemy z współdzieleniem zasobów w środowisku z innymi aplikacjami.

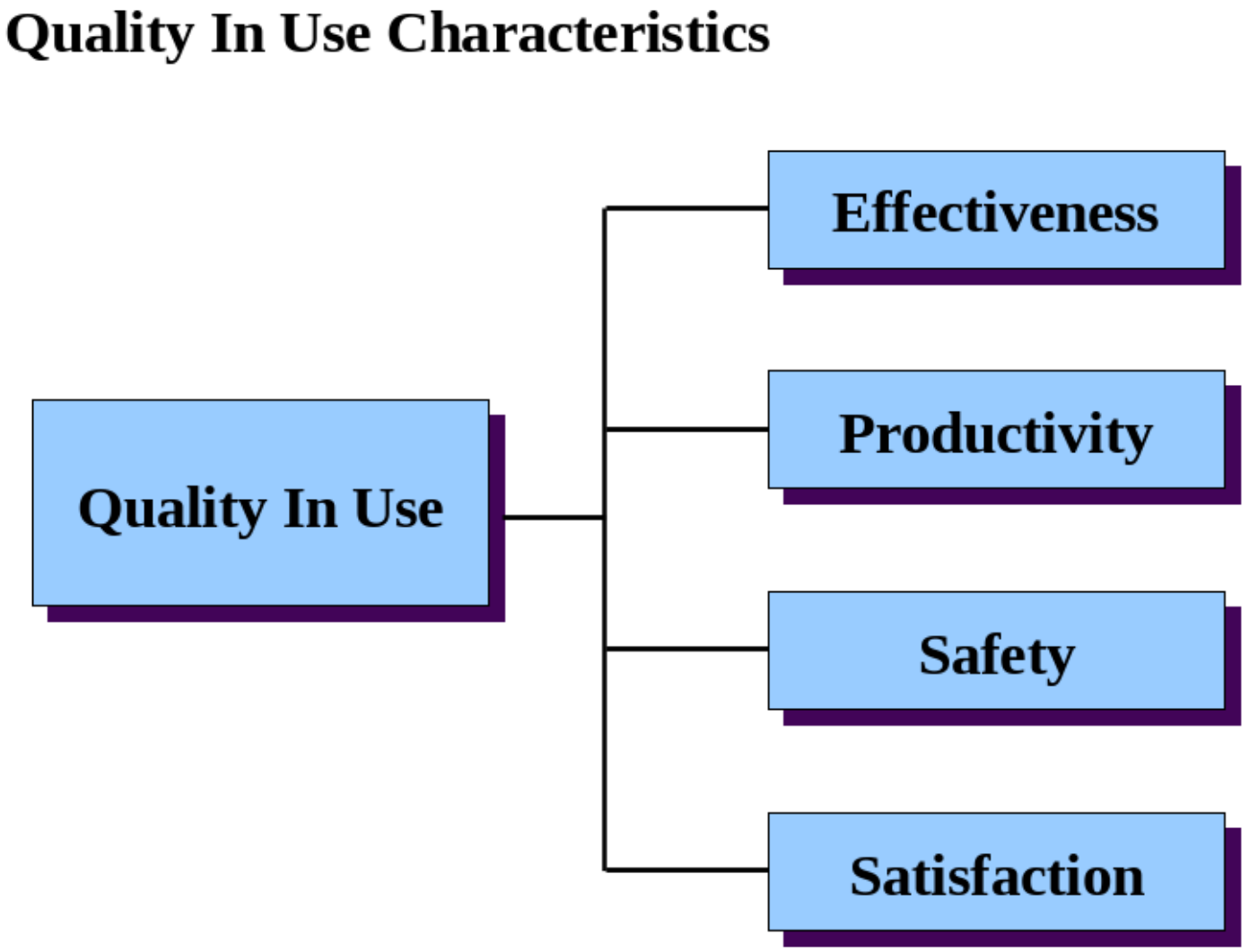
X = A / T
A -> ilość problemów/awarii
T -> czas działania oprogramowania

Przenośność/zastępowanie

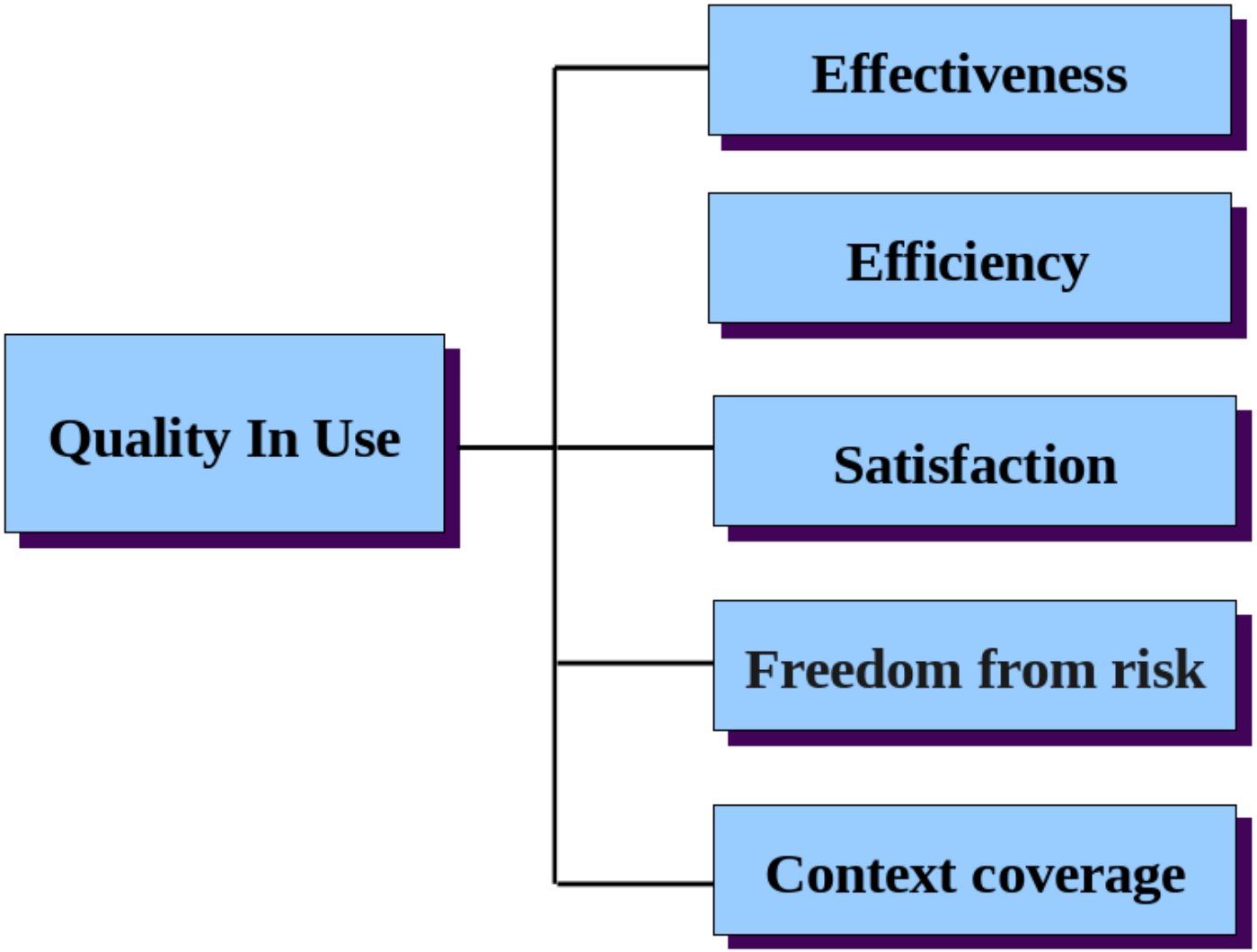
W przypadku podmiany oprogramowania na inne wykonujące to samo zadanie rozpatrujemy następujące kryteria :

- Data continuation
- Function inclusiveness

Model jakości użytkowej (iso9126)



Model jakości użytkowej (SQuaRE)



Skuteczność (Effectiveness)

Ta charakterystyka opisuje w jakim stopniu użytkownik osiągnął zamierzone cele.
Przykładowe miary :

Task completion = (Ukończone zadania)/ (łączna liczba zadań)

Error frequency = (liczba błędów usera)/ (liczba zadań)

Sprawność (Efficiency)

Stosunek wydanych zasobów do osiągniętych efektów. Przykładowe metryki :

- **Time efficiency** = ((Tt = zakładany czas) - (czas faktyczny))/ Tt
- **Relative task time** = (czas zwykłego użytkownika)/(czas przeszkolonego użytkownika)
- **Task efficiency** = (efektywność zadania)/ (czas poświęcony)

Satysfakcja (Satisfaction)

W skład tej charakterystyki wchodzą:

- Usefulness
  - **Proportion of Customer complaints** = A / B
  - **Discretionary utilization of functions** = Σ(Ai) /n
- Trust
  - **Trust scale** (Czy użytkownik ufa systemowi ?)
- Pleasure
  - **Pleasure scale**
- Comfort
  - **Comfort scale**

Freedom from risk

Składowe charakterystyki:

- Risk mitigation
- Financial
  - **Return of investment = A / B**
  - **Delivery time** = (czas faktyczny) / (założenia)
- Health and safety
- Environmental
  - **Environmental impact = (** wpływ faktyczny) / (wpływ założony)

Context coverage

- Context completeness measures
  - **Context completeness**
- Flexibility measures
  - **Flexible context of use**
  - **Flexible design features**