

Inżynieria oprogramowania: narzędzia wspomagające

Zdzisław Sroczyński



Politechnika Śląska
Instytut Matematyki
Wydział Matematyki Stosowanej

▶ metryki i ocena złożoności oprogramowania

- ▶ metryki i ocena złożoności oprogramowania
- ▶ zapewnienie jakości - ISO

- ▶ metryki i ocena złożoności oprogramowania
- ▶ zapewnienie jakości - ISO
- ▶ systemy kontroli wersji i programy instalacyjne

- ▶ metryki i ocena złożoności oprogramowania
- ▶ zapewnienie jakości - ISO
- ▶ systemy kontroli wersji i programy instalacyjne
- ▶ systemy zarządzania projektami (rola dla metodyk zwinnych/agile)

- ▶ metryki i ocena złożoności oprogramowania
- ▶ zapewnienie jakości - ISO
- ▶ systemy kontroli wersji i programy instalacyjne
- ▶ systemy zarządzania projektami (rola dla metodyk zwinnych/agile)

- ▶ metryki i ocena złożoności oprogramowania
- ▶ zapewnienie jakości - ISO
- ▶ systemy kontroli wersji i programy instalacyjne
- ▶ systemy zarządzania projektami (rola dla metodyk zwinnych/agile)

Metryki i ocena złożoności oprogramowania

Metryki i ocena złożoności oprogramowania

- ▶ określenie (przybliżonego) nakładu czasu i pracy wymaganych do wyprodukowania oprogramowania

Metryki i ocena złożoności oprogramowania

- ▶ określenie (przybliżonego) nakładu czasu i pracy wymaganych do wyprodukowania oprogramowania
- ▶ określanie wydajności programistów lub zespołów programistycznych

Metryki i ocena złożoności oprogramowania

- ▶ określenie (przybliżonego) nakładu czasu i pracy wymaganych do wyprodukowania oprogramowania
- ▶ określanie wydajności programistów lub zespołów programistycznych
- ▶ ocena złożoności oprogramowania przed wykonaniem

Metryki i ocena złożoności oprogramowania

- ▶ określenie (przybliżonego) nakładu czasu i pracy wymaganych do wyprodukowania oprogramowania
- ▶ określanie wydajności programistów lub zespołów programistycznych
- ▶ ocena złożoności oprogramowania przed wykonaniem
 - wykorzystanie specyfikacji wymagań

Metryki i ocena złożoności oprogramowania

- ▶ określenie (przybliżonego) nakładu czasu i pracy wymaganych do wyprodukowania oprogramowania
- ▶ określanie wydajności programistów lub zespołów programistycznych
- ▶ ocena złożoności oprogramowania przed wykonaniem
 - wykorzystanie specyfikacji wymagań
 - istotne doświadczenie osoby oceniającej

Metryki i ocena złożoności oprogramowania

- ▶ określenie (przybliżonego) nakładu czasu i pracy wymaganych do wyprodukowania oprogramowania
- ▶ określanie wydajności programistów lub zespołów programistycznych
- ▶ ocena złożoności oprogramowania przed wykonaniem
 - wykorzystanie specyfikacji wymagań
 - istotne doświadczenie osoby oceniającej
 - możliwe porównania z analogicznymi, zakończonymi już projektami

Metryki i ocena złożoności oprogramowania

- ▶ określenie (przybliżonego) nakładu czasu i pracy wymaganych do wyprodukowania oprogramowania
- ▶ określanie wydajności programistów lub zespołów programistycznych
- ▶ ocena złożoności oprogramowania przed wykonaniem
 - wykorzystanie specyfikacji wymagań
 - istotne doświadczenie osoby oceniającej
 - możliwe porównania z analogicznymi, zakończonymi już projektami

Metryki i ocena złożoności oprogramowania

- ▶ określenie (przybliżonego) nakładu czasu i pracy wymaganych do wyprodukowania oprogramowania
- ▶ określanie wydajności programistów lub zespołów programistycznych
- ▶ ocena złożoności oprogramowania przed wykonaniem
 - wykorzystanie specyfikacji wymagań
 - istotne doświadczenie osoby oceniającej
 - możliwe porównania z analogicznymi, zakończonymi już projektami

Metryki i ocena złożoności oprogramowania

Metryki i ocena złożoności oprogramowania

- ▶ ocena złożoności oprogramowania po wykonaniu/zakodowaniu

Metryki i ocena złożoności oprogramowania

- ▶ ocena złożoności oprogramowania po wykonaniu/zakodowaniu
 - programometria oparta na analizie kodu źródłowego

Metryki i ocena złożoności oprogramowania

- ▶ ocena złożoności oprogramowania po wykonaniu/zakodowaniu
 - programometria oparta na analizie kodu źródłowego
 - metody matematyczne i statystyczne

Metryki i ocena złożoności oprogramowania

- ▶ ocena złożoności oprogramowania po wykonaniu/zakodowaniu
 - programometria oparta na analizie kodu źródłowego
 - metody matematyczne i statystyczne
 - uniezależnienie od systemu operacyjnego/języka programowania/stylu i preferencji programisty

Metryki i ocena złożoności oprogramowania

- ▶ ocena złożoności oprogramowania po wykonaniu/zakodowaniu
 - programometria oparta na analizie kodu źródłowego
 - metody matematyczne i statystyczne
 - uniezależnienie od systemu operacyjnego/języka programowania/stylu i preferencji programisty
 - uniezależnienie od potencjalnych refaktoryzacji kodu

Metryki i ocena złożoności oprogramowania

- ▶ ocena złożoności oprogramowania po wykonaniu/zakodowaniu
 - programometria oparta na analizie kodu źródłowego
 - metody matematyczne i statystyczne
 - uniezależnienie od systemu operacyjnego/języka programowania/stylu i preferencji programisty
 - uniezależnienie od potencjalnych refaktoryzacji kodu

- ▶ miary dla programowania strukturalnego

Metryki i ocena złożoności oprogramowania

- ▶ ocena złożoności oprogramowania po wykonaniu/zakodowaniu
 - programometria oparta na analizie kodu źródłowego
 - metody matematyczne i statystyczne
 - uniezależnienie od systemu operacyjnego/języka programowania/stylu i preferencji programisty
 - uniezależnienie od potencjalnych refaktoryzacji kodu

- ▶ miary dla programowania strukturalnego

- ▶ miary dla programowania obiektowego

Metryki i ocena złożoności oprogramowania

- ▶ ocena złożoności oprogramowania po wykonaniu/zakodowaniu
 - programometria oparta na analizie kodu źródłowego
 - metody matematyczne i statystyczne
 - uniezależnienie od systemu operacyjnego/języka programowania/stylu i preferencji programisty
 - uniezależnienie od potencjalnych refaktoryzacji kodu
- ▶ miary dla programowania strukturalnego
- ▶ miary dla programowania obiektowego
- ▶ miary dla projektów bazodanowych/gier?

Metryki i ocena złożoności oprogramowania

- ▶ ocena złożoności oprogramowania po wykonaniu/zakodowaniu
 - programometria oparta na analizie kodu źródłowego
 - metody matematyczne i statystyczne
 - uniezależnienie od systemu operacyjnego/języka programowania/stylu i preferencji programisty
 - uniezależnienie od potencjalnych refaktoryzacji kodu
- ▶ miary dla programowania strukturalnego
- ▶ miary dla programowania obiektowego
- ▶ miary dla projektów bazodanowych/gier?

Metryki i ocena złożoności oprogramowania

- ▶ ocena złożoności oprogramowania po wykonaniu/zakodowaniu
 - programometria oparta na analizie kodu źródłowego
 - metody matematyczne i statystyczne
 - uniezależnienie od systemu operacyjnego/języka programowania/stylu i preferencji programisty
 - uniezależnienie od potencjalnych refaktoryzacji kodu
- ▶ miary dla programowania strukturalnego
- ▶ miary dla programowania obiektowego
- ▶ miary dla projektów bazodanowych/gier?

Miary dla programowania strukturalnego:

Miary dla programowania strukturalnego:

- ▶ wg rozmiaru programu

Miary dla programowania strukturalnego:

- ▶ wg rozmiaru programu
 - NCL (ang. *Number of Code Lines* = LOC (ang. *Lines Of Code*) - liczba (niepustych) wierszy kodu - parametr zależny od stylu

Miary dla programowania strukturalnego:

▶ wg rozmiaru programu

- NCL (ang. *Number of Code Lines* = LOC (ang. *Lines Of Code*) - liczba (niepustych) wierszy kodu - parametr zależny od stylu
- NI (ang. *Number of Instructions* - liczba instrukcji - parametr niezależny od stylu, zależny od języka programowania

Miary dla programowania strukturalnego:

- ▶ wg rozmiaru programu
 - NCL (ang. *Number of Code Lines* = LOC (ang. *Lines Of Code*) - liczba (niepustych) wierszy kodu - parametr zależny od stylu
 - NI (ang. *Number of Instructions* - liczba instrukcji - parametr niezależny od stylu, zależny od języka programowania
- ▶ wg struktury programu:

Miary dla programowania strukturalnego:

- ▶ wg rozmiaru programu
 - NCL (ang. *Number of Code Lines* = LOC (ang. *Lines Of Code*) - liczba (niepustych) wierszy kodu - parametr zależny od stylu
 - NI (ang. *Number of Instructions* - liczba instrukcji - parametr niezależny od stylu, zależny od języka programowania
- ▶ wg struktury programu:
 - struktura modułów, np. liczba CYCLO=liczba instrukcji warunkowych i pętli-1, względna złożoność logiczna=liczba instrukcji decyzyjnych/liczba instrukcji

Miary dla programowania strukturalnego:

▶ wg rozmiaru programu

- NCL (ang. *Number of Code Lines* = LOC (ang. *Lines Of Code*) - liczba (niepustych) wierszy kodu - parametr zależny od stylu
- NI (ang. *Number of Instructions* - liczba instrukcji - parametr niezależny od stylu, zależny od języka programowania

▶ wg struktury programu:

- struktura modułów, np. liczba CYCLO=liczba instrukcji warunkowych i pętli-1, względna złożoność logiczna=liczba instrukcji decyzyjnych/liczba instrukcji
- struktura programu, np. złożoność strukturalna C_S=liczba odwołań do zewnętrznych modułów/liczba modułów

Miary dla programowania strukturalnego:

▶ wg rozmiaru programu

- NCL (ang. *Number of Code Lines* = LOC (ang. *Lines Of Code*) - liczba (niepustych) wierszy kodu - parametr zależny od stylu
- NI (ang. *Number of Instructions* - liczba instrukcji - parametr niezależny od stylu, zależny od języka programowania

▶ wg struktury programu:

- struktura modułów, np. liczba CYCLO=liczba instrukcji warunkowych i pętli-1, względna złożoność logiczna=liczba instrukcji decyzyjnych/liczba instrukcji
- struktura programu, np. złożoność strukturalna C_S=liczba odwołań do zewnętrznych modułów/liczba modułów
- sugestywność, np. VAR=średnia długość nazw

Miary dla programowania strukturalnego:

▶ wg rozmiaru programu

- NCL (ang. *Number of Code Lines* = LOC (ang. *Lines Of Code*) - liczba (niepustych) wierszy kodu - parametr zależny od stylu
- NI (ang. *Number of Instructions* - liczba instrukcji - parametr niezależny od stylu, zależny od języka programowania

▶ wg struktury programu:

- struktura modułów, np. liczba CYCLO=liczba instrukcji warunkowych i pętli-1, względna złożoność logiczna=liczba instrukcji decyzyjnych/liczba instrukcji
- struktura programu, np. złożoność strukturalna C_S=liczba odwołań do zewnętrznych modułów/liczba modułów
- sugestywność, np. VAR=średnia długość nazw

Miary dla programowania strukturalnego:

▶ wg rozmiaru programu

- NCL (ang. *Number of Code Lines* = LOC (ang. *Lines Of Code*) - liczba (niepustych) wierszy kodu - parametr zależny od stylu
- NI (ang. *Number of Instructions* - liczba instrukcji - parametr niezależny od stylu, zależny od języka programowania

▶ wg struktury programu:

- struktura modułów, np. liczba CYCLO=liczba instrukcji warunkowych i pętli-1, względna złożoność logiczna=liczba instrukcji decyzyjnych/liczba instrukcji
- struktura programu, np. złożoność strukturalna C_S=liczba odwołań do zewnętrznych modułów/liczba modułów
- sugestywność, np. VAR=średnia długość nazw

Miary dla programowania strukturalnego wg struktury programu:

Miary dla programowania strukturalnego wg struktury programu:

- ▶ złożoność operacyjna

$$R=c1*VAR + c2*NCL + c3*CYCLO$$

Miary dla programowania strukturalnego wg struktury programu:

- ▶ złożoność operacyjna

$$R=c1*VAR + c2*NCL + c3*CYCLO$$

- ▶ Metryka Halsteada:

Miary dla programowania strukturalnego wg struktury programu:

- ▶ złożoność operacyjna

$$R = c1 * VAR + c2 * NCL + c3 * CYCLO$$

- ▶ Metryka Halsteada:

$$difficulty D = \frac{liczba_unikatowych_operatorow}{2} \frac{liczba_wystapien_operandow}{liczba_unikatowych_operandow}$$

Miary dla programowania strukturalnego wg struktury programu:

- ▶ złożoność operacyjna

$$R = c1 * VAR + c2 * NCL + c3 * CYCLO$$

- ▶ Metryka Halsteada:

$$difficulty D = \frac{liczba_unikatowych_operatorow}{2} \frac{liczba_wystapien_operandow}{liczba_unikatowych_operandow}$$

$$volume V = liczba_wystapien_operatorow_i_operandow * \log_2(liczba_operatorow + liczba_operandow)$$

Miary dla programowania strukturalnego wg struktury programu:

- ▶ złożoność operacyjna

$$R = c1 * VAR + c2 * NCL + c3 * CYCLO$$

- ▶ Metryka Halsteada:

$$difficulty D = \frac{liczba_unikatowych_operatorow}{2} \frac{liczba_wystapien_operandow}{liczba_unikatowych_operandow}$$

$$volume V = liczba_wystapien_operatorow_i_operandow * \log_2(liczba_operatorow + liczba_operandow)$$

$$effort E = V * D$$

Miary dla programowania strukturalnego wg struktury programu:

- ▶ złożoność operacyjna

$$R = c1 * VAR + c2 * NCL + c3 * CYCLO$$

- ▶ Metryka Halsteada:

$$difficulty D = \frac{liczba_unikatowych_operatorow}{2} \frac{liczba_wystapien_operandow}{liczba_unikatowych_operandow}$$

$$volume V = liczba_wystapien_operatorow_i_operandow * \log_2(liczba_operatorow + liczba_operandow)$$

$$effort E = V * D$$

Miary dla programowania strukturalnego wg struktury programu:

- ▶ złożoność operacyjna

$$R = c1 * VAR + c2 * NCL + c3 * CYCLO$$

- ▶ Metryka Halsteada:

$$difficulty D = \frac{liczba_unikatowych_operatorow}{2} \frac{liczba_wystapien_operandow}{liczba_unikatowych_operandow}$$

$$volume V = liczba_wystapien_operatorow_i_operandow * \log_2(liczba_operatorow + liczba_operandow)$$

$$effort E = V * D$$

Miary dla programowania obiektowego:

Miary dla programowania obiektowego:

- ▶ proste (metod, klas, systemu):

Miary dla programowania obiektowego:

- ▶ proste (metod, klas, systemu):
 - NCLM (ang. *Number of Code Lines in Method* - liczba (niepustych) wierszy kodu w metodzie

Miary dla programowania obiektowego:

- ▶ proste (metod, klas, systemu):
 - NCLM (ang. *Number of Code Lines in Method* - liczba (niepustych) wierszy kodu w metodzie
 - NIM (ang. *Number of Instructions in Method* - liczba instrukcji w metodzie)

Miary dla programowania obiektowego:

▶ proste (metod, klas, systemu):

- NCLM (ang. *Number of Code Lines in Method* - liczba (niepustych) wierszy kodu w metodzie
- NIM (ang. *Number of Instructions in Method* - liczba instrukcji w metodzie
- średni rozmiar metody $\text{avg}(\text{NCLM}) < 18$

Miary dla programowania obiektowego:

- ▶ proste (metod, klas, systemu):
 - NCLM (ang. *Number of Code Lines in Method* - liczba (niepustych) wierszy kodu w metodzie
 - NIM (ang. *Number of Instructions in Method* - liczba instrukcji w metodzie
 - średni rozmiar metody $\text{avg}(\text{NCLM}) < 18$
 - N_{Meth} - liczba metod w klasie

Miary dla programowania obiektowego:

▶ proste (metod, klas, systemu):

- NCLM (ang. *Number of Code Lines in Method* - liczba (niepustych) wierszy kodu w metodzie
- NIM (ang. *Number of Instructions in Method* - liczba instrukcji w metodzie
- średni rozmiar metody $\text{avg}(\text{NCLM}) < 18$
- N_{Meth} - liczba metod w klasie
- $N_{PublMeth}$ - liczba metod publicznych w klasie

Miary dla programowania obiektowego:

▶ proste (metod, klas, systemu):

- NCLM (ang. *Number of Code Lines in Method* - liczba (niepustych) wierszy kodu w metodzie
- NIM (ang. *Number of Instructions in Method* - liczba instrukcji w metodzie
- średni rozmiar metody $\text{avg}(\text{NCLM}) < 18$
- N_{Meth} - liczba metod w klasie
- $N_{PublMeth}$ - liczba metod publicznych w klasie
- ... liczba metod statycznych, dziedziczonych, polimorficznych

Miary dla programowania obiektowego:

▶ proste (metod, klas, systemu):

- NCLM (ang. *Number of Code Lines in Method* - liczba (niepustych) wierszy kodu w metodzie
- NIM (ang. *Number of Instructions in Method* - liczba instrukcji w metodzie
- średni rozmiar metody $\text{avg}(\text{NCLM}) < 18$
- N_{Meth} - liczba metod w klasie
- $N_{PublMeth}$ - liczba metod publicznych w klasie
- ... liczba metod statycznych, odziedziczonych, polimorficznych
- analogicznie j/w ale dla pól

Miary dla programowania obiektowego:

▶ proste (metod, klas, systemu):

- NCLM (ang. *Number of Code Lines in Method* - liczba (niepustych) wierszy kodu w metodzie
- NIM (ang. *Number of Instructions in Method* - liczba instrukcji w metodzie
- średni rozmiar metody $\text{avg}(\text{NCLM}) < 18$
- N_{Meth} - liczba metod w klasie
- $N_{PublMeth}$ - liczba metod publicznych w klasie
- ... liczba metod statycznych, odziedziczonych, polimorficznych
- analogicznie j/w ale dla pól
- N_{Class} - liczba klas w programie

Miary dla programowania obiektowego:

▶ proste (metod, klas, systemu):

- NCLM (ang. *Number of Code Lines in Method* - liczba (niepustych) wierszy kodu w metodzie
- NIM (ang. *Number of Instructions in Method* - liczba instrukcji w metodzie
- średni rozmiar metody $\text{avg}(\text{NCLM}) < 18$
- N_{Meth} - liczba metod w klasie
- $N_{PublMeth}$ - liczba metod publicznych w klasie
- ... liczba metod statycznych, odziedziczonych, polimorficznych
- analogicznie j/w ale dla pól
- N_{Class} - liczba klas w programie
- $N_{AbstrClass} = 0, 1 * N_{Class}$ - liczba klas abstrakcyjnych

Miary dla programowania obiektowego:

▶ proste (metod, klas, systemu):

- NCLM (ang. *Number of Code Lines in Method* - liczba (niepustych) wierszy kodu w metodzie
- NIM (ang. *Number of Instructions in Method* - liczba instrukcji w metodzie
- średni rozmiar metody $\text{avg}(\text{NCLM}) < 18$
- N_{Meth} - liczba metod w klasie
- $N_{PublMeth}$ - liczba metod publicznych w klasie
- ... liczba metod statycznych, odziedziczonych, polimorficznych
- analogicznie j/w ale dla pól
- N_{Class} - liczba klas w programie
- $N_{AbstrClass} = 0, 1 * N_{Class}$ - liczba klas abstrakcyjnych

Miary dla programowania obiektowego:

Miary dla programowania obiektowego:

▶ złożone:

Miary dla programowania obiektowego:

▶ złożone:

- $N_{AvgMeth} < 12 - 25$ - średnia liczba metod w klasie

Miary dla programowania obiektowego:

▶ złożone:

- $N_{AvgMeth} < 12 - 25$ - średnia liczba metod w klasie
- $N_{AvgAttr} < 3 - 9$ - średnia liczba pól w klasie

Miary dla programowania obiektowego:

▶ złożone:

- $N_{AvgMeth} < 12 - 25$ - średnia liczba metod w klasie
- $N_{AvgAttr} < 3 - 9$ - średnia liczba pól w klasie
- MHF (ang. *Method Hiding Factor*) - współczynnik hermetyzacji metod

Miary dla programowania obiektowego:

▶ złożone:

- $N_{AvgMeth} < 12 - 25$ - średnia liczba metod w klasie
- $N_{AvgAttr} < 3 - 9$ - średnia liczba pól w klasie
- MHF (ang. *Method Hiding Factor*) - współczynnik hermetyzacji metod
- AHF (ang. *Attribute Hiding Factor*) - współczynnik hermetyzacji pól

Miary dla programowania obiektowego:

▶ złożone:

- $N_{AvgMeth} < 12 - 25$ - średnia liczba metod w klasie
- $N_{AvgAttr} < 3 - 9$ - średnia liczba pól w klasie
- MHF (ang. *Method Hiding Factor*) - współczynnik hermetyzacji metod
- AHF (ang. *Attribute Hiding Factor*) - współczynnik hermetyzacji pól
- MIF (ang. *Method Inheritance Factor*) - współczynnik dziedziczenia metod

Miary dla programowania obiektowego:

▶ złożone:

- $N_{AvgMeth} < 12 - 25$ - średnia liczba metod w klasie
- $N_{AvgAttr} < 3 - 9$ - średnia liczba pól w klasie
- MHF (ang. *Method Hiding Factor*) - współczynnik hermetyzacji metod
- AHF (ang. *Attribute Hiding Factor*) - współczynnik hermetyzacji pól
- MIF (ang. *Method Inheritance Factor*) - współczynnik dziedziczenia metod
- AIF (ang. *Attribute Inheritance Factor*) - współczynnik dziedziczenia pól

Miary dla programowania obiektowego:

▶ złożone:

- $N_{AvgMeth} < 12 - 25$ - średnia liczba metod w klasie
- $N_{AvgAttr} < 3 - 9$ - średnia liczba pól w klasie
- MHF (ang. *Method Hiding Factor*) - współczynnik hermetyzacji metod
- AHF (ang. *Attribute Hiding Factor*) - współczynnik hermetyzacji pól
- MIF (ang. *Method Inheritance Factor*) - współczynnik dziedziczenia metod
- AIF (ang. *Attribute Inheritance Factor*) - współczynnik dziedziczenia pól
- MPF (ang. *Method Polymorphism Factor*) - współczynnik polimorfizmu metod

Miary dla programowania obiektowego:

▶ złożone:

- $N_{AvgMeth} < 12 - 25$ - średnia liczba metod w klasie
- $N_{AvgAttr} < 3 - 9$ - średnia liczba pól w klasie
- MHF (ang. *Method Hiding Factor*) - współczynnik hermetyzacji metod
- AHF (ang. *Attribute Hiding Factor*) - współczynnik hermetyzacji pól
- MIF (ang. *Method Inheritance Factor*) - współczynnik dziedziczenia metod
- AIF (ang. *Attribute Inheritance Factor*) - współczynnik dziedziczenia pól
- MPF (ang. *Method Polymorphism Factor*) - współczynnik polimorfizmu metod

Systemy zapewnienia jakości w zakresie IT:

ISO (International Organization for Standardization)
IEC (International Electrotechnical Commission)

Systemy zapewnienia jakości w zakresie IT:

ISO (International Organization for Standardization)
IEC (International Electrotechnical Commission)



Systemy zapewnienia jakości:

Systemy zapewnienia jakości:

- ▶ ISO 9001:2008 to norma międzynarodowa zawierająca wymagania dotyczące systemów zarządzania jakością w organizacjach potrzebujących wykazać zdolność do ciągłego dostarczania wyrobu spełniającego wymagania klienta oraz wymagania w zakresie nadrzędnych przepisów

Systemy zapewnienia jakości:

- ▶ ISO 9001:2008 to norma międzynarodowa zawierająca wymagania dotyczące systemów zarządzania jakością w organizacjach potrzebujących wykazać zdolność do ciągłego dostarczania wyrobu spełniającego wymagania klienta oraz wymagania w zakresie nadrzędnych przepisów
- ▶ ISO 20000 Zarządzanie usługami IT: standard efektywnego zarządzania działaniem pionów IT, zarówno własnych, jak i do usług świadczonych dla klientów zewnętrznych, np. dostarczanie usług teleinformatycznych, outsourcing informatyczny

Systemy zapewnienia jakości:

- ▶ ISO 9001:2008 to norma międzynarodowa zawierająca wymagania dotyczące systemów zarządzania jakością w organizacjach potrzebujących wykazać zdolność do ciągłego dostarczania wyrobu spełniającego wymagania klienta oraz wymagania w zakresie nadrzędnych przepisów
- ▶ ISO 20000 Zarządzanie usługami IT: standard efektywnego zarządzania działaniem pionów IT, zarówno własnych, jak i do usług świadczonych dla klientów zewnętrznych, np. dostarczanie usług teleinformatycznych, outsourcing informatyczny

ISO 9001:

ISO 9001:

- ▶ norma opisuje działania, które muszą wykonać dostawca i nabywca oprogramowania w trakcie całego cyklu życia systemu

ISO 9001:

- ▶ norma opisuje działania, które muszą wykonać dostawca i nabywca oprogramowania w trakcie całego cyklu życia systemu
- ▶ Plan zarządzania jakością

ISO 9001:

- ▶ norma opisuje działania, które muszą wykonać dostawca i nabywca oprogramowania w trakcie całego cyklu życia systemu
- ▶ Plan zarządzania jakością
 - mierzalne cele jakościowe

ISO 9001:

- ▶ norma opisuje działania, które muszą wykonać dostawca i nabywca oprogramowania w trakcie całego cyklu życia systemu
- ▶ Plan zarządzania jakością
 - mierzalne cele jakościowe
 - kryteria dla wejść i wyjść z każdego etapu pracy

ISO 9001:

- ▶ norma opisuje działania, które muszą wykonać dostawca i nabywca oprogramowania w trakcie całego cyklu życia systemu
- ▶ Plan zarządzania jakością
 - mierzalne cele jakościowe
 - kryteria dla wejść i wyjść z każdego etapu pracy
 - opis testów, weryfikacji i sposobów ich zatwierdzania

ISO 9001:

- ▶ norma opisuje działania, które muszą wykonać dostawca i nabywca oprogramowania w trakcie całego cyklu życia systemu
- ▶ Plan zarządzania jakością
 - mierzalne cele jakościowe
 - kryteria dla wejść i wyjść z każdego etapu pracy
 - opis testów, weryfikacji i sposobów ich zatwierdzania
 - plany związane z testami i weryfikacjami oraz wskazanie osób upoważnionych do ich zatwierdzania

ISO 9001:

- ▶ norma opisuje działania, które muszą wykonać dostawca i nabywca oprogramowania w trakcie całego cyklu życia systemu
- ▶ Plan zarządzania jakością
 - mierzalne cele jakościowe
 - kryteria dla wejść i wyjść z każdego etapu pracy
 - opis testów, weryfikacji i sposobów ich zatwierdzania
 - plany związane z testami i weryfikacjami oraz wskazanie osób upoważnionych do ich zatwierdzania
 - określenie odpowiedzialności za działania związane z jakością takie jak testy i przeglądy, zarządzanie konfiguracją, zarządzanie zmianami, postępowanie z wyrobami nie spełniającymi wymagań

ISO 9001:

- ▶ norma opisuje działania, które muszą wykonać dostawca i nabywca oprogramowania w trakcie całego cyklu życia systemu
- ▶ Plan zarządzania jakością
 - mierzalne cele jakościowe
 - kryteria dla wejść i wyjść z każdego etapu pracy
 - opis testów, weryfikacji i sposobów ich zatwierdzania
 - plany związane z testami i weryfikacjami oraz wskazanie osób upoważnionych do ich zatwierdzania
 - określenie odpowiedzialności za działania związane z jakością takie jak testy i przeglądy, zarządzanie konfiguracją, zarządzanie zmianami, postępowanie z wyrobami nie spełniającymi wymagań

ISO 9126 - jakość oprogramowania:

ISO 9126 - jakość oprogramowania:

- ▶ funkcjonalność

ISO 9126 - jakość oprogramowania:

- ▶ funkcjonalność
- ▶ niezawodność

ISO 9126 - jakość oprogramowania:

- ▶ funkcjonalność
- ▶ niezawodność
- ▶ użyteczność

ISO 9126 - jakość oprogramowania:

- ▶ funkcjonalność
- ▶ niezawodność
- ▶ użyteczność
- ▶ efektywność

ISO 9126 - jakość oprogramowania:

- ▶ funkcjonalność
- ▶ niezawodność
- ▶ użyteczność
- ▶ efektywność
- ▶ pielęgnowalność

ISO 9126 - jakość oprogramowania:

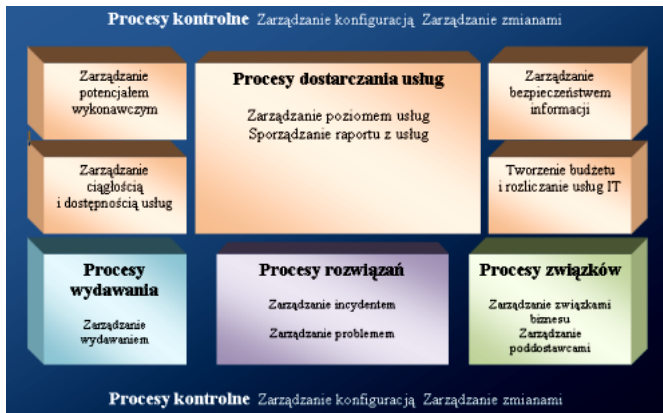
- ▶ funkcjonalność
- ▶ niezawodność
- ▶ użyteczność
- ▶ efektywność
- ▶ pielęgnowalność
- ▶ przenaszalność

ISO 9126 - jakość oprogramowania:

- ▶ funkcjonalność
- ▶ niezawodność
- ▶ użyteczność
- ▶ efektywność
- ▶ pielęgnowalność
- ▶ przenaszalność

ISO/IEC 20000-1:2005

ISO/IEC 20000-1:2005



O certyfikację ISO/IEC 20000-1:2005 mogą ubiegać się organizacje:

O certyfikację ISO/IEC 20000-1:2005 mogą ubiegać się organizacje:

- ▶ zamierzające świadczyć usługi IT

O certyfikację ISO/IEC 20000-1:2005 mogą ubiegać się organizacje:

- ▶ zamierzające świadczyć usługi IT
- ▶ wymagające spójnego podejścia wszystkich dostawców usług w łańcuchu dostawców

O certyfikację ISO/IEC 20000-1:2005 mogą ubiegać się organizacje:

- ▶ zamierzające świadczyć usługi IT
- ▶ wymagające spójnego podejścia wszystkich dostawców usług w łańcuchu dostawców
- ▶ chcące wykazać zdolność do zapewnienia usług, które spełniają wymagania klienta

O certyfikację ISO/IEC 20000-1:2005 mogą ubiegać się organizacje:

- ▶ zamierzające świadczyć usługi IT
- ▶ wymagające spójnego podejścia wszystkich dostawców usług w łańcuchu dostawców
- ▶ chcące wykazać zdolność do zapewnienia usług, które spełniają wymagania klienta
- ▶ dążące do ciągłego doskonalenie usługi poprzez procesy monitorowania i doskonalenia jakości usług

O certyfikację ISO/IEC 20000-1:2005 mogą ubiegać się organizacje:

- ▶ zamierzające świadczyć usługi IT
- ▶ wymagające spójnego podejścia wszystkich dostawców usług w łańcuchu dostawców
- ▶ chcące wykazać zdolność do zapewnienia usług, które spełniają wymagania klienta
- ▶ dążące do ciągłego doskonalenie usługi poprzez procesy monitorowania i doskonalenia jakości usług
- ▶ dla których dział IT/usługi IT są bardzo ważnym elementem biznesu

O certyfikację ISO/IEC 20000-1:2005 mogą ubiegać się organizacje:

- ▶ zamierzające świadczyć usługi IT
- ▶ wymagające spójnego podejścia wszystkich dostawców usług w łańcuchu dostawców
- ▶ chcące wykazać zdolność do zapewnienia usług, które spełniają wymagania klienta
- ▶ dążące do ciągłego doskonalenie usługi poprzez procesy monitorowania i doskonalenia jakości usług
- ▶ dla których dział IT/usługi IT są bardzo ważnym elementem biznesu

Norma ISO 27001:

Norma ISO 27001:

- ▶ Wymagania dla systemu zarządzania bezpieczeństwem informacji (SZBI). System ten w swojej budowie i działaniu jest zbliżony do systemu zarządzania jakością według normy ISO 9001

Norma ISO 27001:

- ▶ Wymagania dla systemu zarządzania bezpieczeństwem informacji (SZBI). System ten w swojej budowie i działaniu jest zbliżony do systemu zarządzania jakością według normy ISO 9001
- ▶ Poufność - zapewnienie, że informacje są dostępne tylko dla osób uprawnionych do ich dostępu

Norma ISO 27001:

- ▶ Wymagania dla systemu zarządzania bezpieczeństwem informacji (SZBI). System ten w swojej budowie i działaniu jest zbliżony do systemu zarządzania jakością według normy ISO 9001
- ▶ Poufność - zapewnienie, że informacje są dostępne tylko dla osób uprawnionych do ich dostępu
- ▶ Integralność – zagwarantowanie dokładności i kompletności informacji oraz metod ich przetwarzania

Norma ISO 27001:

- ▶ Wymagania dla systemu zarządzania bezpieczeństwem informacji (SZBI). System ten w swojej budowie i działaniu jest zbliżony do systemu zarządzania jakością według normy ISO 9001
- ▶ Poufność - zapewnienie, że informacje są dostępne tylko dla osób uprawnionych do ich dostępu
- ▶ Integralność – zagwarantowanie dokładności i kompletności informacji oraz metod ich przetwarzania
- ▶ Dostępność – zapewnienie upoważnionym użytkownikom dostępu do informacji i związanych z nimi zasobów, zgodnie z określonymi potrzebami

Norma ISO 27001:

- ▶ Wymagania dla systemu zarządzania bezpieczeństwem informacji (SZBI). System ten w swojej budowie i działaniu jest zbliżony do systemu zarządzania jakością według normy ISO 9001
- ▶ Poufność - zapewnienie, że informacje są dostępne tylko dla osób uprawnionych do ich dostępu
- ▶ Integralność – zagwarantowanie dokładności i kompletności informacji oraz metod ich przetwarzania
- ▶ Dostępność – zapewnienie upoważnionym użytkownikom dostępu do informacji i związanych z nimi zasobów, zgodnie z określonymi potrzebami

Systemy kontroli wersji:

Systemy kontroli wersji:

- ▶ CVS (ang. *Concurrent Versions System*) - licencja GPL

Systemy kontroli wersji:

- ▶ CVS (ang. *Concurrent Versions System*) - licencja GPL
- ▶ SVN (ang. *Subversion*) - licencja Apache

Systemy kontroli wersji:

- ▶ CVS (ang. *Concurrent Versions System*) - licencja GPL
- ▶ SVN (ang. *Subversion*) - licencja Apache
 - historia zmian plików

Systemy kontroli wersji:

- ▶ CVS (ang. *Concurrent Versions System*) - licencja GPL
- ▶ SVN (ang. *Subversion*) - licencja Apache
 - historia zmian plików
 - atomowe potwierdzenia zmian

Systemy kontroli wersji:

- ▶ CVS (ang. *Concurrent Versions System*) - licencja GPL
- ▶ SVN (ang. *Subversion*) - licencja Apache
 - historia zmian plików
 - atomowe potwierdzenia zmian
 - efektywne odgałęzienia

Systemy kontroli wersji:

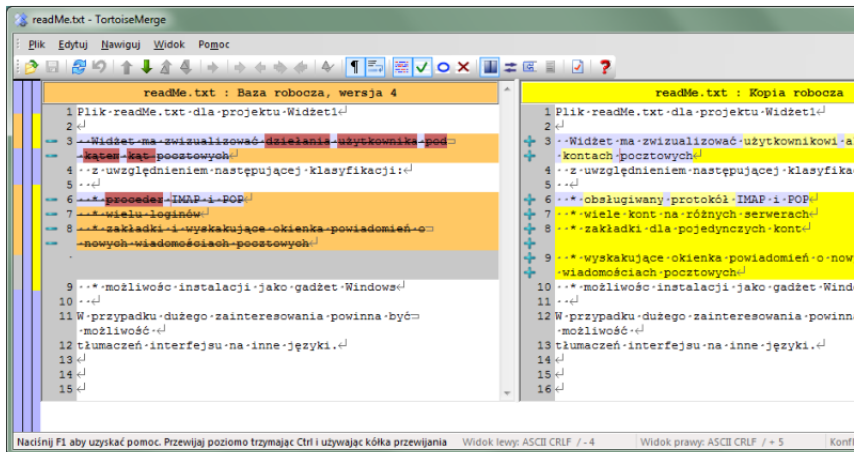
- ▶ CVS (ang. *Concurrent Versions System*) - licencja GPL
- ▶ SVN (ang. *Subversion*) - licencja Apache
 - historia zmian plików
 - atomowe potwierdzenia zmian
 - efektywne odgałęzienia
 - możliwość użycia dedykowanego serwera lub serwera Apache z WebDAV

Systemy kontroli wersji:

- ▶ CVS (ang. *Concurrent Versions System*) - licencja GPL
- ▶ SVN (ang. *Subversion*) - licencja Apache
 - historia zmian plików
 - atomowe potwierdzenia zmian
 - efektywne odgałęzienia
 - możliwość użycia dedykowanego serwera lub serwera Apache z WebDAV
 - integracja z systemem operacyjnym, IDE programistycznymi (np. TortoiseSVN)

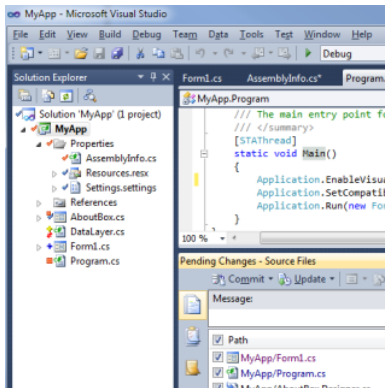
TortoiseSVN

TortoiseSVN



AnkhSVN (Visual Studio, licencja Apache)

AnkhSVN (Visual Studio, licencja Apache)



RAD Studio - historia plików

RAD Studio - historia plików

Unit1

Unit1.pas

Differences From:

| Revision | Date |
|----------|---------------------|
| File | 2013-01-03 13:54:54 |
| Buffer | 2013-01-03 13:54:54 |
| ~29~ | 2013-01-02 14:52:40 |
| ~28~ | 2008-11-28 09:58:32 |
| ~27~ | 2008-11-27 16:14:16 |
| ~26~ | 2008-11-27 10:55:20 |
| ~25~ | 2008-11-13 16:27:06 |
| ~24~ | 2008-11-13 16:12:38 |
| ~23~ | 2008-11-13 16:11:22 |
| ~22~ | 2008-11-13 16:10:20 |
| ~21~ | 2008-11-13 16:07:40 |
| ~20~ | 2008-11-13 16:06:50 |

To:

| Revision | Date |
|----------|---------------------|
| ~29~ | 2013-01-02 14:52:40 |
| ~28~ | 2008-11-28 09:58:32 |
| ~27~ | 2008-11-27 16:14:16 |
| ~26~ | 2008-11-27 10:55:20 |
| ~25~ | 2008-11-13 16:27:06 |
| ~24~ | 2008-11-13 16:12:38 |
| ~23~ | 2008-11-13 16:11:22 |
| ~22~ | 2008-11-13 16:10:20 |
| ~21~ | 2008-11-13 16:07:40 |
| ~20~ | 2008-11-13 16:06:50 |

```
100 tekst:array[1..100] of char;
    bufor:array[1..100] of AnsiChar; //Nie Char!!!
    licznik, ilewczytano: integer;
begin
    licznik:=0;
    if OpenDialog1.Execute then
    begin
        Strumien := TFileStream.Create(OpenDialog1.FileName, fmOpenRead);
        Strumien.Read(tekst, 5);
        Memo1.Lines.Add(tekst);
        while licznik<Strumien.Size do
        begin
110     ilewczytano:=Strumien.Read(bufor, 100);
            Memo1.Text:=Memo1.Text+Copy(bufor, 0, ilewczytano);
            licznik:=licznik+ilewczytano;
        end;
        Strumien.Free;
```

Programy instalacyjne dla Windows (InnoSetup):

Programy instalacyjne dla Windows (InnoSetup):

- ▶ obsługa różnych wersji systemu operacyjnego

Programy instalacyjne dla Windows (InnoSetup):

- ▶ obsługa różnych wersji systemu operacyjnego
- ▶ wiele wersji instalacyjnych, w tym językowych

Programy instalacyjne dla Windows (InnoSetup):

- ▶ obsługa różnych wersji systemu operacyjnego
- ▶ wiele wersji instalacyjnych, w tym językowych
- ▶ automatyczne dołączanie niezbędnych plików i modułów

Programy instalacyjne dla Windows (InnoSetup):

- ▶ obsługa różnych wersji systemu operacyjnego
- ▶ wiele wersji instalacyjnych, w tym językowych
- ▶ automatyczne dołączanie niezbędnych plików i modułów
- ▶ program deinstalacyjny

Programy instalacyjne dla Windows (InnoSetup):

- ▶ obsługa różnych wersji systemu operacyjnego
- ▶ wiele wersji instalacyjnych, w tym językowych
- ▶ automatyczne dołączanie niezbędnych plików i modułów
- ▶ program deinstalacyjny
- ▶ kompresja

Programy instalacyjne dla Windows (InnoSetup):

- ▶ obsługa różnych wersji systemu operacyjnego
- ▶ wiele wersji instalacyjnych, w tym językowych
- ▶ automatyczne dołączanie niezbędnych plików i modułów
- ▶ program deinstalacyjny
- ▶ kompresja
- ▶ tworzenie skrótów, wpisów do rejestru i plików ini

Programy instalacyjne dla Windows (InnoSetup):

- ▶ obsługa różnych wersji systemu operacyjnego
- ▶ wiele wersji instalacyjnych, w tym językowych
- ▶ automatyczne dołączanie niezbędnych plików i modułów
- ▶ program deinstalacyjny
- ▶ kompresja
- ▶ tworzenie skrótów, wpisów do rejestru i plików ini
- ▶ uruchamianie zainstalowanej aplikacji

Programy instalacyjne dla Windows (InnoSetup):

- ▶ obsługa różnych wersji systemu operacyjnego
- ▶ wiele wersji instalacyjnych, w tym językowych
- ▶ automatyczne dołączanie niezbędnych plików i modułów
- ▶ program deinstalacyjny
- ▶ kompresja
- ▶ tworzenie skrótów, wpisów do rejestru i plików ini
- ▶ uruchamianie zainstalowanej aplikacji
- ▶ kodowanie, ochrona hasłem, podpisy cyfrowe dla wersji instalacyjnej

Programy instalacyjne dla Windows (InnoSetup):

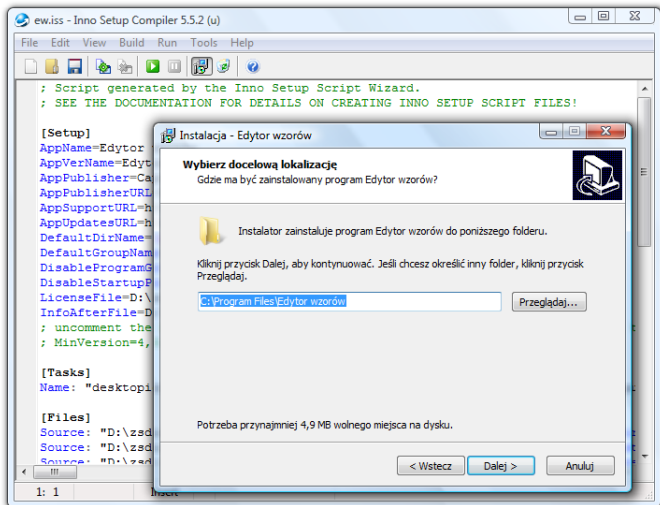
- ▶ obsługa różnych wersji systemu operacyjnego
- ▶ wiele wersji instalacyjnych, w tym językowych
- ▶ automatyczne dołączanie niezbędnych plików i modułów
- ▶ program deinstalacyjny
- ▶ kompresja
- ▶ tworzenie skrótów, wpisów do rejestru i plików ini
- ▶ uruchamianie zainstalowanej aplikacji
- ▶ kodowanie, ochrona hasłem, podpisy cyfrowe dla wersji instalacyjnej
- ▶ obsługa własnych skryptów

Programy instalacyjne dla Windows (InnoSetup):

- ▶ obsługa różnych wersji systemu operacyjnego
- ▶ wiele wersji instalacyjnych, w tym językowych
- ▶ automatyczne dołączanie niezbędnych plików i modułów
- ▶ program deinstalacyjny
- ▶ kompresja
- ▶ tworzenie skrótów, wpisów do rejestru i plików ini
- ▶ uruchamianie zainstalowanej aplikacji
- ▶ kodowanie, ochrona hasłem, podpisy cyfrowe dla wersji instalacyjnej
- ▶ obsługa własnych skryptów

Innosetup:

Innosetup:



Systemy zarządzania projektami - przegląd

Część przykładów zaczerpnięta z materiałów
TUV Nord Polska, IMS MALON oraz SYBENA.

Dziękuję za uwagę

Następny temat:
PREZENTACJA PROJEKTÓW