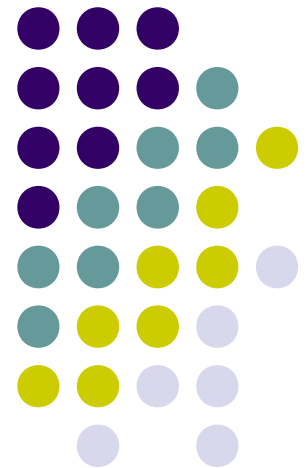


Średniozaawansowane programowanie w C++

Wykład #9
15 grudnia 2016 r.



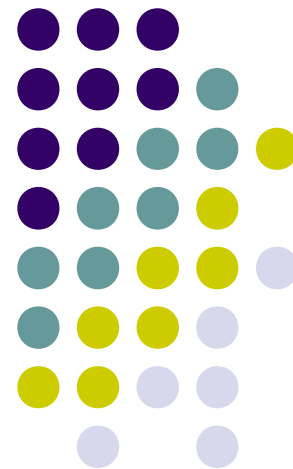
Metodyki wytwarzania oprogramowania

O praktyce mówimy wtedy, gdy system działa,
choć nikt nie wie dlaczego.

O teorii mówimy, gdy system informatyczny
nie działa, ale znamy powód.

My łączymy teorię z praktyką:

nic nie działa i nikt nie wie dlaczego.



Etapy realizacji projektu



To co klient zamówił



To co analityk zrozumiał



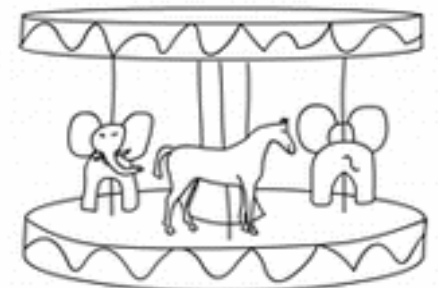
To co opisywał



To co zrobili programiści

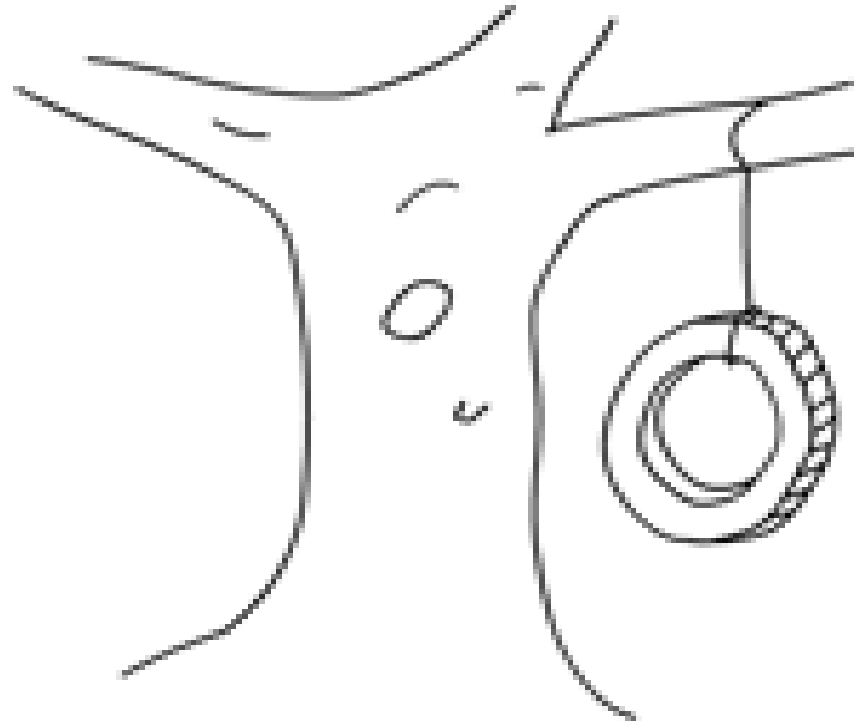
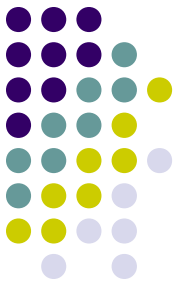


To co powstało po wdrożeniu



To za co klient zapłacił

Etapy realizacji projektu



A to, czego potrzebował

Metodyki zarządzania projektami



Metodyki lekkie

nie mamy teczek, ale mamy dobre chęci i otwarte głowy

eXtreme Programming (XP)

Scrum

Future Driven Development (FDD)

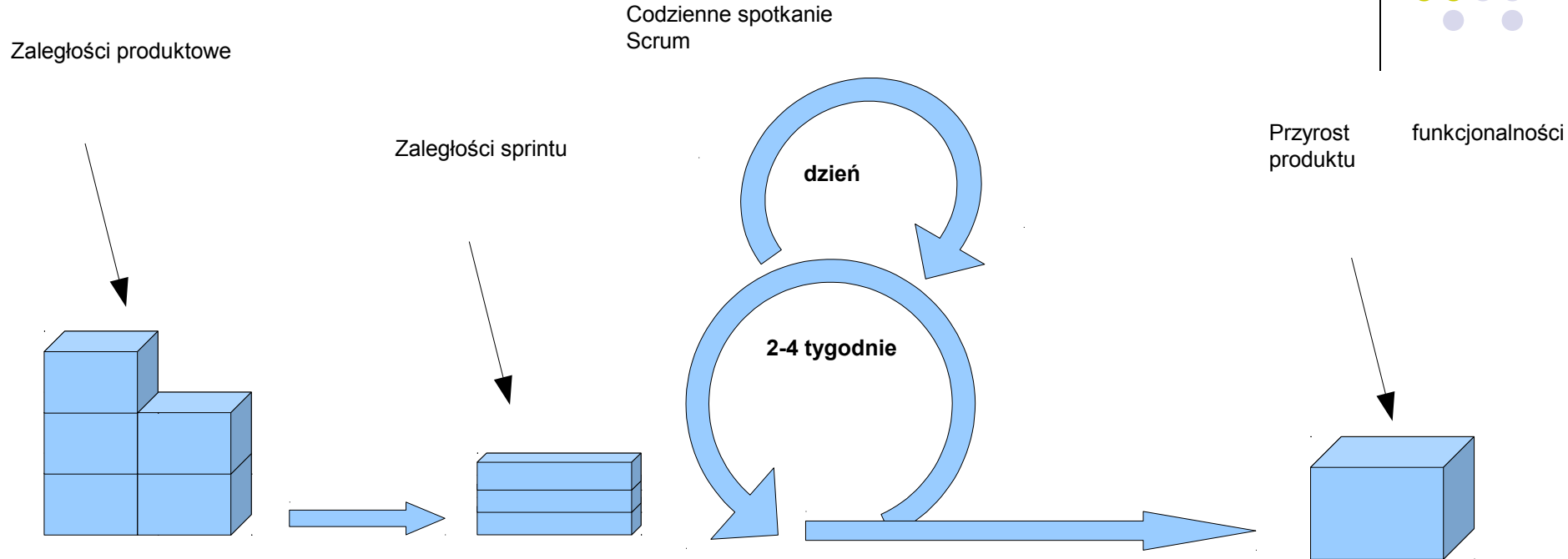
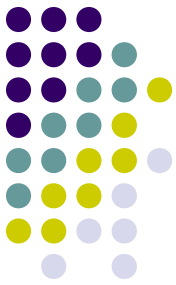
Merodyki ciężkie

jesteśmy wielcy i wszystko trzymamy w teczkach

Prince2

PMI PMBok

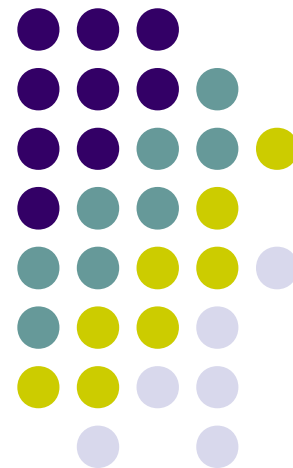
Scrum



1. Dostarczanie kolejnych, coraz bardziej dopracowanych wyników projektu
2. Włączanie się przyszłych użytkowników w proces twórczy
3. Samoorganizacja zespołu projektowego:
 - a) "Mistrz Młyna" (Scrum Master)
 - b) Właściciel Produktu (Product Owner)
 - c) Członkowie Zespołu (The Team)

Paczka

Zawartość paczki



Zawartość paczki – program *Kaczka*



Plik **kaczka-1.0-rc1.tar.gz**

kaczka-1.0-rc1/**src**/ – kod źródłowy (*.cpp, *.hpp)

kaczka-1.0-rc1/**tmp**/ – miejsce na kompilowane pliki *.o

kaczka-1.0-rc1/**bin**/ – katalog na zbudowane binarki

kaczka-1.0-rc1/**doc**/ – dokumentacja programisty

kaczka-1.0-rc1/**man**/ – dokumentacja użytkownika

kaczka-1.0-rc1/**Makefile** – makefile :)

kaczka-1.0-rc1/**AUTHORS** – autor(zy) programu

Budowa paczki tar.gz

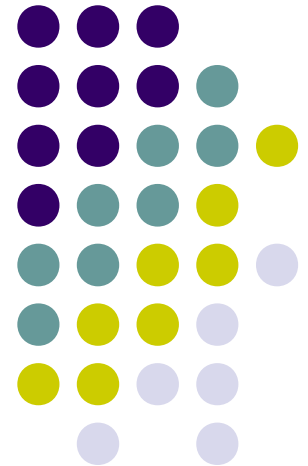


W katalogu nadrzędnym wywołujemy instrukcję:

```
tar -c kaczka-1.0-rc1 | gzip -9 > kaczka-1.0-rc1.tar.gz
```

Dokumentowanie kodu

Doxygen



Mainpage



```
/**
 * @mainpage
 *      Program wyświetlający obrazy radarowe i pozwalający użytkownikom na
zaznaczanie obiektu na ekranie
 *      @author Tomasz Karol Pietrzak
 *      @date    2010.01.12
 *      @version      1.0-rc1
 */
```

Opis klasy – plik *.hpp



```
/**
 * Proxy do klasy Radarogram.
 * Decyduje o wczytywaniu obrazow radarowych, przechowuje połozenie obiektu
 * oraz dostarcza metod zwalnajacych przydzielona pamiec
 */
class RadarogramPRX
{
    public:
        explicit RadarogramPRX (std::string nazwa);
        ~RadarogramPRX ();
        Radarogram *operator-> ();
        BITMAP *bmp (const Paleta *pal);
        std::string nazwa () const;
        void ustawP0 (const int2 &po);
        std::pair <int, int> pobierzP0 () const;
        void zwolnij ();

    private:
        Radarogram *rdg_;
        BITMAP *rdg_bmp_;           // bitmapa wygenerowana dla danej palety
        std::string nazwa_;         // nazwa pliku
        const Paleta *pal_;        // paleta uzyta do wygenerowania bitmapy
        int2 po_;                  // połozenie zaznaczonego obiektu
        boost::mutex mtx_;

};
```

Opis funkcji/metody – plik *.cpp



```
/**
 * Zwraca referencje do wybranego radarogramu.
 * Jesli nastepuje przekroczenie zakresu, jest rzucany wyjatek
 * @param i numer indeksu wybieranego radarogramu
 * @return Referencja do wybranego radarogramu
 */
RadarogramPRX &BuforPamieci::r (int i)
{
    if (i < 0) i = i_;
    if (i >= static_cast<int>(ilosc())) throw BładPrzekroczonyZakres ();
    kol_dostepu_ [i] = licznik_;
    licznik_++;
    return *vrp_ [i];
}
```

Generowanie dokumentacji



```
# Tworzymy plik konfiguracyjny
doxygen -g tmp/dokumentacja.conf

# Edytujemy plik konfiguracyjny (jeśli potrzeba)
vim tmp/dokumentacja.conf

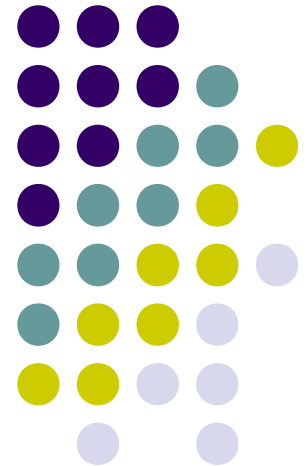
# Np.:
PROJECT_NAME      = „Moj program”
OUTPUT_LANGUAGE   = Polish

# W katalogu głównym programu odpalamy...
doxygen tmp/dokumentacja.conf

# Zostaną utworzone katalogi:
# html – dokumentacja w HTML
# latex – dokumentacja w latexie z makefilem do zrobienia PDF
```

Dodatki

Deasemblacja
Benchmarki



Deasemblacja



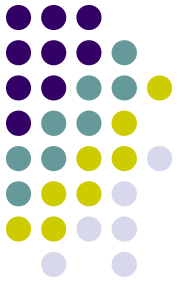
1. GCC

g++ -S kod.cpp (...)

2. On-line

<https://godbolt.org>

Google::benchmark



<https://github.com/google/benchmark>

Programowanie jest fantastyczne!!!

