



Akademia ADB

**Wykład I – Praca w grupie**

# O wykładach...

O czym jest ta seria wykładów?

- Wstęp do pracy w grupie programistów nad projektem informatycznym.
- Przeznaczenie: studenci chcący być programistami i nie tylko

Czym nie są te wykłady:

- Kompendium
- Rozważaniami teoretycznymi
- Ostateczną wykładnią





# Kto pracuje nad projektem?

Nad projektem software'owym pracuje wiele osób o różnych funkcjach (interesariusze, *stakeholders*):

- Przedstawiciele klienta
- Kierownicy projektu (*project managers*)
- Analitycy
- Architekci
- Programiści
- Testerzy (grupy QC)



# Fazy projektu informatycznego

- Planowanie (specyfikacja wymagań – *requirements*)
  - Analiza systemu – czyli opis działania
  - Projekt – czyli architektura systemu
  - Implementacja
  - Testowanie
  - Wdrożenie i utrzymanie (*maintenance*)
- ... to model kaskadowy.

Prosty, zrozumiały i nie do odtworzenia w świecie rzeczywistym.





# Metodyka Agile - teoria

Agile to zestaw metod, które prowadzą do realizacji projektu programistycznego. Podstawa teoretyczna to *Agile Manifesto* z 2001 roku:

- Ludzie i ich współpraca są ważniejsze niż narzędzia i procesy
- Działające oprogramowanie jest ważniejsze niż dokumentacja
- Istotna jest współpraca z klientem, a nie negocjacje
- Plan nie jest świętością – liczy się odpowiedź na zmiany



# Metodyka Agile - praktyka

W praktyce, manifest realizowany jest poprzez:

- iteracyjną metody pracy nad małymi etapami,
- małe grup skupiające ludzi o różnych funkcjach,
- klucz to zmotywowani ludzie, organizujący się sami i obdarzeni zaufaniem
- częste dostarczanie działającego oprogramowania,
- ciągłą obecnością przedstawiciela klienta,
- użycie jasnego obrazowania postępu prac,
- częste, nieformalne spotkania grupy pozwalające na natychmiastową reakcję na problemy
- wolny dobór narzędzi – zasada **it depends**



# Poznawanie wymagań

- Wymagania powinny zostać określone w trakcie długich rozmów z klientem.
- Mówimy językiem klienta.
- Klient może odkryć swoje potrzeby w trakcie definiowania systemu.
- Żądaną funkcjonalność rozbijamy na opowieści użytkownika (*user stories*)
- Dążymy do prostoty opowieści i ich weryfikowalności
- Opowieści grupujemy w epiki (*epics*)
- Opowieści są podstawą szacowania pracy





## Przykład fiszki z historią użytkownika

Jako administrator, mogę wskazać pliki do codziennego sporządzania kopii zapasowej.



# JIRA – narzędzie Agile

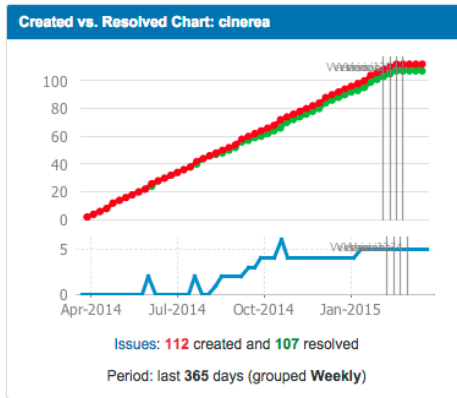
New

+ Add Gadget Edit Layout ⚙️ Tools

**Two Dimensional Filter Statistics: All Issues**

Status	Assignee				
	Admin Istrator	Atvmlyhh Uottoson	Dzoqurfs Udbbigyh	Ipxyfkjt Khjwqeee	Ltkq Urpa
OPEN	7	4	3	5	2
IN PROGRESS	16	14	10	11	16
REOPENED	9	11	10	10	17
RESOLVED	34	22	19	15	18
CLOSED	20	19	18	15	11

Showing 5 of 10 statistics. [Show more](#)  
Filter: All issues

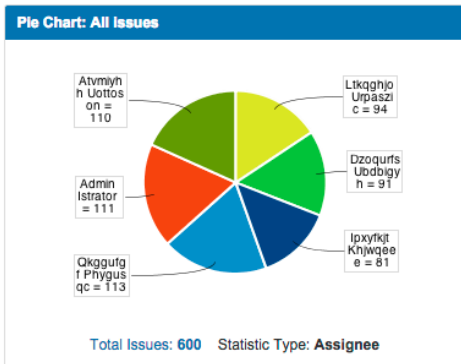


**Issue Statistics**

Statistics: All Issues (Priority)

Blocker	2	9%
Critical	6	26%
Major	5	22%
Minor	4	17%
Trivial	6	26%

Total Issues: 23



**Assigned to Me**

T	Key	P	Summary
📄	GGIS-9	↑	GGIS-7 / Satined snot greyhound's demolish Savonarola's naphthalene's premium's Humberto's earache's
📄	LICS-41	↑	Handball's cued ruggedly bacchanalians resonance occasions resonators uncompromising
📄	CNEA-90	↑	Redolence silencers fortifications payload's demolitions Chicagoan comports Kristine Gregory dockyard
📄	GGIS-4	↓	Hypnotism incinerating jasper's zip's equipage's Glass Lester's trollops enchanting coxswains
📄	CNEA-97	↓	Extracurriculars falsehood Indian overreacts manicurists

1-5 of 5

**Filter Results: All Issues**

T	Key	P	Summary
📄	TMBI-121	↑	Messiest gratuity's convalescents afloat Amway's
📄	TMBI-120	↑	Passageway's prefab's clunk begone brainchild's vireo commentator's plover
📄	TMBI-119	↓	Ptomaine hences attentions blacked diocese grimed brazier's hippopotami
📄	TMBI-118	↑	TMBI-116 / Pineapple Frobisher's buzzers raisins clothing
📄	TMBI-117	↓	TMBI-116 / Majesties Carissa Sammy mooting lonesomes bourgeois

1-5 of 600 1 2 3 4 5 6 7 8 9 ▶



# Issue – podstawa pracy z JIRA

The screenshot shows a JIRA issue page for 'Pomodoro Web / POM-11'. The issue title is 'As a user, I would like to be able to remove items from the To Do Today list so that I can plan properly'. The issue is currently in 'In Progress' status. The page includes sections for Details, Description, Sub-Tasks, Activity, People, Dates, Time Tracking, and Agile. The Sub-Tasks section lists two tasks: 'Implement REST endpoint' and 'Wire up backbone and invoke model destroy from link/button', both assigned to 'admin' and showing 0% progress. The Agile section shows the issue is currently in 'Sprint 4 - Bolognese' which ends on 09/Aug/12.

**JIRA** admin Administration

Dashboards Projects Issues Agile + Create Issue Quick Search

Pomodoro Web / POM-11 2 of 23 Return to search

## As a user, I would like to be able to remove items from the To Do Today list so that I can plan properly

Edit Assign Comment More Actions Stop Progress Resolve Issue Workflow Views

### Details

Type: Story  
Priority: Minor  
Affects Version/s: None  
Labels: None

Status: In Progress (View Workflow)  
Resolution: Unresolved  
Fix Version/s: None

### People

Assignee: admin  
Reporter: admin  
Vote (0) Watch (0)

### Dates

Created: 30/Apr/12 2:37 PM  
Updated: Today 3:10 PM

### Time Tracking

Estimated: 2d 4h  
Remaining: 1d 4h  
Logged: Not Specified  
 Include sub-tasks

### Agile

Active Sprint: Sprint 4 - Bolognese ends 09/Aug/12  
Completed Sprint: Sprint 3 - Arrabiata ended 23/Jul/12  
[View on Board](#)

### Description

Acceptance criteria:

- It should be possible to remove an item with one click
- There shouldn't be a confirmation, store the item in a "trash" location of some sort so it can be restored

### Sub-Tasks

Task	Status	Assignee	Progress
1. Implement REST endpoint	Reopened	admin	0%
2. Wire up backbone and invoke model destroy from link/button	Open	admin	0%

### Activity

All Comments Work Log History Activity

There are no comments yet on this issue.

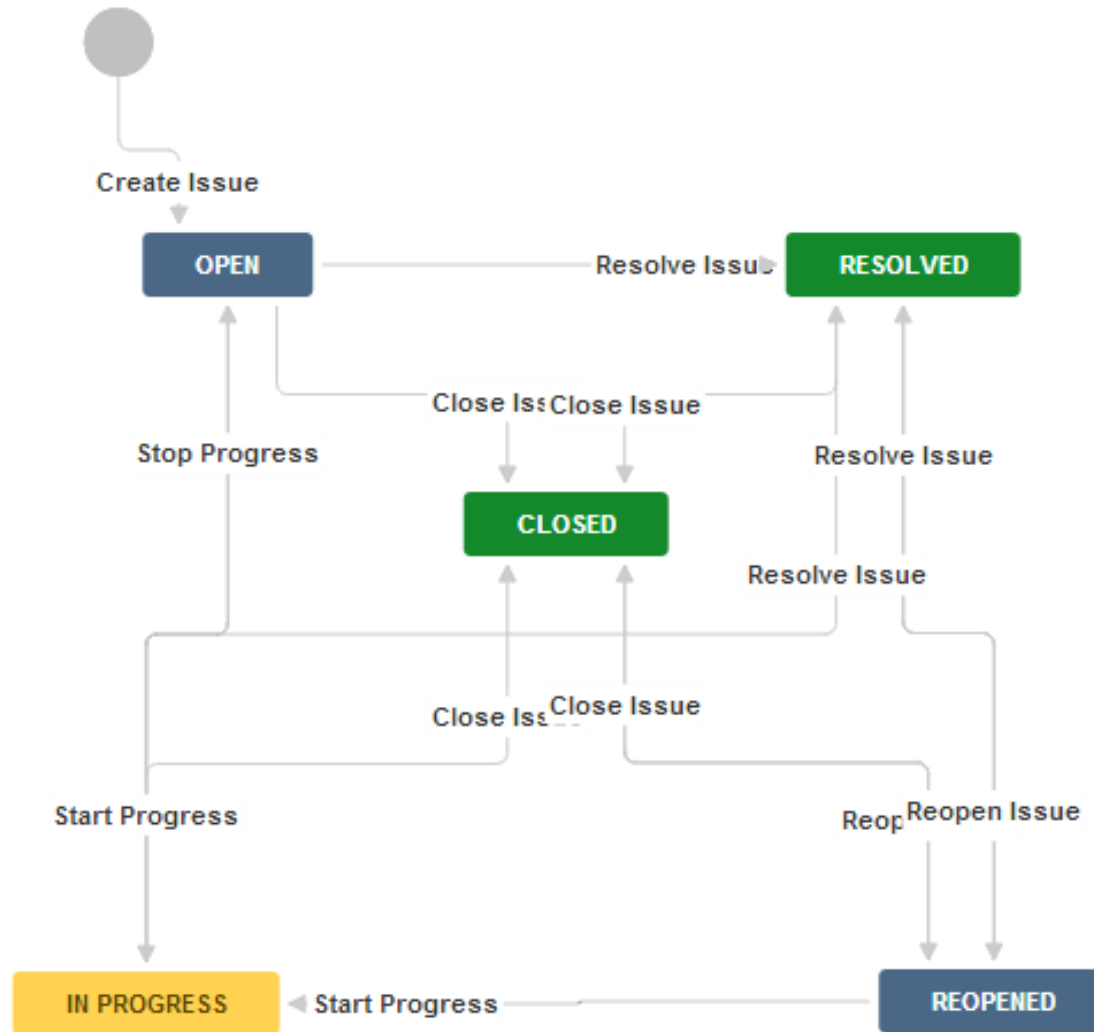
Comment

Bug tracking and project tracking for software development powered by Atlassian JIRA (v5.1#768-sha1:38ad524) | Report a problem





# Cykl życia issue



Sample Scrum Project / SSP-38

## Estimates epic

Edit Comment Assign More ▾ To Do In Progress Done Admin ▾ Export ▾

### Details

Type:	🚩 Epic	Status:	<b>TO DO</b> (View Workflow)
Priority:	↓ Minor	Resolution:	Unresolved
Affects Version/s:	None	Fix Version/s:	None
Component/s:	None		
Labels:	None		
Epic Name:	Estimates		

### Description

[Click to add description](#)

### Issues in Epic

SSP-2	As a product owner, I'd like to express work in terms of actual user problems, aka User Stories, and place them in the backlog	🚩	<b>TO DO</b>	Andrew Lui
SSP-5	As a team, I'd like to commit to a set of stories to be completed in a sprint (or iteration)	🚩	<b>TO DO</b>	Unassigned
SSP-4	As a team, I'd like to estimate the effort of a story in Story Points so we can understand the work remaining	🚩	<b>TO DO</b>	Sarah Maddox
SSP-9	As a developer, I'd like to update story status during the sprint	🚩	<b>TO DO</b>	Unassigned

### People

Assignee: Unassigned  
[Assign to me](#)

Reporter: Andrew Lui

Votes:

Watchers: [Stop watching this issue](#)

### Dates

Created: 03/Sep/14 2:07 PM

Updated: 1 minute ago

### Development

[Create branch](#)

### Drag and Drop

Drop files here to attach them  
or  
[Select files](#)

# Projekty

## Sample Project

Key: SP | URL: No URL | Category: None

[Edit Project](#) [Actions](#)

- Summary
- Issue Types
- Workflows
- Screens
- Fields
- Roles
- Permissions
- Issue Security
- Notifications
- Issue Collectors

### Issue Types

Keep track of different types of issues, such as bugs or tasks. Each issue type can be configured differently.

Scheme: [Default Issue Type Scheme](#)

- Bug
- Improvement
- New Feature
- Task
- Sub-task (Sub-Task)

[More](#)

### Workflows

Issues can follow processes that mirror your team's practices. A workflow defines the sequence of steps that an issue will follow, e.g. "In Progress", "Resolved".

Scheme: [Default Workflow Scheme](#)

[jira](#)

[More](#)

### Screens

Screens allow you to arrange the fields to be displayed for an issue. Different screens can be used when an issue is created, viewed, edited, or transitioned through a workflow.

Scheme: [Default Issue Type Screen Scheme](#)

- [Default Screen Scheme \(Default\)](#)

[More](#)

### Fields

Different issues can have different information fields. A field configuration defines how fields behave for the project, e.g. required/optional; hidden/visible.

Scheme: [System Default Field Configuration](#)

- [Default Field Configuration \(Default\)](#)

[More](#)

### Settings

Some general project configuration options.

CVS Modules: [None \(Change\)](#)

### Versions

For software projects, JIRA allows you to track different versions, e.g. 1.0, 2.0. Issues can be assigned to versions.

This project has no unarchived versions. [Add a version](#)

### Components

Projects can be broken down into components, e.g. "Database", "User Interface". Issues can then be categorised against different components.

This project does not use any components. [Add a component](#)

### Roles

JIRA enables you to allocate particular people to specific roles in your project. Roles are used when defining other settings, like notifications and permissions.

Project Lead: [charlie of atlassian](#)

Default Assignee: [Project Lead](#)

Roles: [View Project Roles](#)

[More](#)

### Permissions

Project permissions allow you to control who can access your project, and what they can do, e.g. "Work on Issues". Access to individual issues is granted to people by issue permissions.

Scheme: [Default Permission Scheme](#)

Issues: [None](#)

### Notifications

JIRA can notify the appropriate people of particular events in your project, e.g. "Issue Commented". You can choose specific people, groups, or roles to receive notifications.

Scheme: [None](#)

Email: [No mail server configured](#)

[More](#)





# Scrum

Technika agile służąca do organizowania pracy grupy programistycznej:

- Pracę gromadzi się w backlogu
- Iteracje są krótkie i o ustalonej długości
- Na końcu iteracji dostępny jest działający produkt oddawany klientowi do oceny



# Scrum - zalety

Główne zalety stosowania Scrum to:

- niższe koszty
- lepszy zwrot inwestycji
- wyniki są widoczne i to bardzo szybko
- większa pewność co do stanu projektu

= zadowolenie klienta



# Scrum - role

W scrumie pojawiają się następujące funkcje:

- właściciel produktu
- Scrum Master
- wielofunkcyjny, autonomiczny zespół



# Scrum – cykl pracy

W ramach pojedynczej iteracji (sprintu) występują etapy:

- planowanie
- codzienny scrum: mniej niż kwadrans
- praca
- przegląd (*review*): przegląd + prezentacja oprogramowania
- Retrospektywa: refleksja + poprawa
- scrum nad scrumami (*Scrum of Scrums*): spotkanie koordynujące ambasadorów kilku grup



# Scrum – Sprint

Cechy dobrego sprintu:

- jasno definiuje co robimy;
- ogranicza ilość zadań, nad którymi pracujemy (*WIP – work in progress*);
- jest krótki i niezmienny (zwykle 2 tyg. lub miesiąc);
- jeśli zadania nie zostaną wykonane, wracają do backlogu



# Scrum – Backlog

Cechy backlogu:

- praca do wykonania (usterki, nowe funkcje, prace infrastrukturalne, zdobywanie wiedzy itp.);
- zawiera szczegółowe elementy;
- backlog jest płynny;
- elementy mają określony priorytet i koszt wykonania.





# JIRA – Backlog

## Backlog

Board ▾ ⬆

🔍 QUICK FILTERS: [Only My Issues](#) [Recently Updated](#)

**Sample Sprint 5** 4 issues 8 0 3

02/Sep/14 5:34 PM • 15/Sep/14 5:34 PM Linked pages

**EPICS**

- ↑ **SSP-4** As a team, I'd like to estimate the effort of a story in Story Points so we can understand the Version 3.0 **Estimates** 5
- ↑ **SSP-2** As a product owner, I'd like to express work in terms of actual user problems, aka User Story Version 2.0 **Estimates** 2
- ↑ **SSP-6** As a scrum master, I'd like to break stories down into tasks we can track during the sprint Version 2.0 **Estimates** 1
- ↑ **SSP-14** As a user, I can find important items on the board by using the customisable "Quick Filters" ab Version 2.0 **Filters** 3

**Backlog** 9 issues Create Sprint

- ↑ **SSP-1** As an Agile team, I'd like to learn about Scrum Version 2.0 2
- ↑ **SSP-8** As a product owner, I'd like to include bugs, tasks and other issue types in my backlog Version 2.0 1
- ↑ **SSP-5** As a team, I'd like to commit to a set of stories to be completed in a sprint (or iteration) Version 2.0 **Estimates** 1

### Sample Scrum Project / SSP-4

As a team, I'd like to estimate the effort of a story in Story Points so we can understand the work remaining

Estimate: 5

**Details**

Status: **TO DO**

Component/s: Board

Labels: None

Affects Version/s: None

Fix Version/s: Version 3.0

Epic: **Estimates x**

**People**

# Scrum – Wady

Scrum nie jest wolny od wad. W szczególności problemem są:

- projekty bez określonego celu;
- projekty z bardzo zmiennym zakresem – mniejszym niż czas sprintu;
- wbrew pozorom zarządzanie z punktu widzenia mistrza nie jest proste;
- podobnie jak cała metodologia agile, odczuwany może być brak sformalizowanego podejścia.



# Kanban – pomoc w chaotycznym projekcie

The screenshot shows a JIRA Kanban board for the project 'Scrum: Teams in Space'. The board is organized into four columns: '4 To Do', '4 In Progress (Min 3, Max 5)', '1 Code Review', and '4 Done'. Issues are represented as cards with titles, descriptions, and progress indicators. A sidebar on the right provides details for the selected issue TIS-67, including reporter and assignee information, dates, and development status.

Issue ID	Description	Progress
TIS-46	Update LocalTransportC to handle	6
TIS-40	Update FlightController to handle	6
TIS-6	Requesting available flights is now taking >	0
TIS-69	Add a String anonymizer to TextUtils	0
TIS-45	Email non registered users to sign	2
TIS-49	Draft network plan for Mars Office	5
TIS-67	Developer Toolbox does not display by	0
TIS-19	Engage Saturn Space Tours Group Travel	5
TIS-43	Extend experience in UI to include	9
TIS-68	Homepage footer uses an inline style -	0
TIS-66	Add pointer to main css file to instruct users	0
TIS-44	Reward Customers an extra 5-10%	3
TIS-42	Extend booking experience in UI to include	9

**Issue Details (TIS-67):**

- People:** Reporter: Jennifer Evans, Assignee: Jennifer Evans
- Dates:** Created: 30/Jan/14 3:43 PM, Updated: 03/Feb/14 3:22 PM
- Development:** 5 branches (Updated 13/Feb/14), 3 commits (Latest a day ago), 1 pull request (Updated a day ago), 1 build (Latest 13/Feb/14)



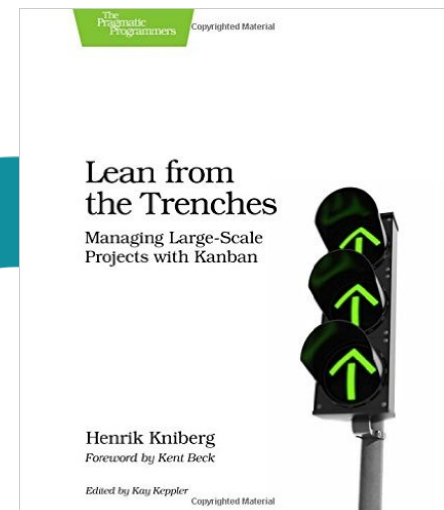
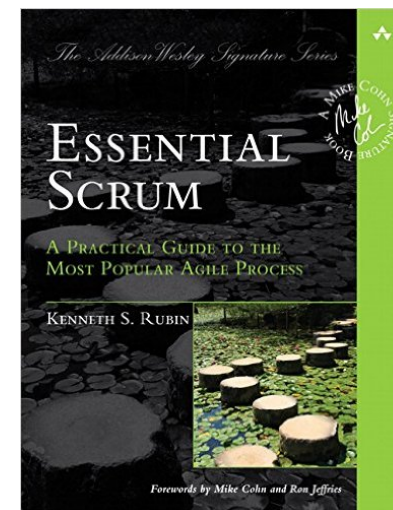
# Podsumowanie

- Projekt, którego celem jest stworzenie oprogramowania, angażuje wiele osób i zawiera wiele elementów.
- Końcowa forma projektu nie jest całkowicie zgodna z początkowymi założeniami.
- W celu sprawnego zarządzania wytwarzaniem oprogramowania stosowane są techniki Agile.
- Techniki Agile to zbiór, z którego można stosować wybrane elementy.
- JIRA Agile świetnie pomaga w organizacji pracy.
- Bardzo popularną metodologią zarządzania tworzeniem oprogramowania jest Scrum.
- Kanban jest prostszą alternatywą dla Scruma.



# Do poczytania...

- R. C. Martin i M. Martin, *Agile. Programowanie zwinne: zasady, wzorce i praktyki zwinnego wytwarzania oprogramowania w C#*
- Strona domowa JIRY: <https://www.atlassian.com/software/jira>
- K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*
- H. Kniberg, *Lean from the Trenches: Managing Large-Scale Projects with Kanban*







Koniec

Dziękuję za uwagę

[adbglobal.com](http://adbglobal.com)