



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Grzegorz Rogus, Piotr Szwed, Jan Werewka

Porównanie metodyk zarządzania projektami PMBOK i Scrum przy użyciu modeli ontologicznych

Część III/III

Laboratorium Informatyki, Katedra Automatyki, AGH, Wydział EAIiE
Seminarium PMI Oddz. Kraków i Explicite KA AGH, 17 czerwca 2010



Spis treści

- Co to jest ontologia?
- Dlaczego tworzymy ontologię?
- Komponenty ontologii
- Proces tworzenia ontologii – krok po kroku





AGH

Agenda

- Ontologie PMBOK i Scrum – informacje ogólne
 - Architektura
 - Jak powstawały ontologie PMBOK i Scrum
- Role
 - Role PMBOK
 - Role Scrum
- Artefakty
 - Artefakty PMBOK
 - Artefakty Scrum
- Zdarzenia
 - Zdarzenia PMBOK
 - Zdarzenia Scrum
- Procesy
 - Procesy PMBOK
 - Procesy Scrum
- Zastosowania
- Podsumowanie



AGH

O ontologiach PMBOK i Scrum



Architektura ontologii PMBOK i Scrum

- Podstawowe elementy składowe to:
 - **Role** (określające funkcje osób lub grup osób w metodyce)
 - **Artefakty** (wejścia, wyjścia, produkty procesów zarządzania)
 - **Zdarzenia** (obiekty o pewnych własnościach czasowych, jak czas wystąpienia lub czas trwania)
 - **Procesy** (opisane przez podanie wejść, wyjść i niezbędnych zasobów: ról, narzędzi i technik)

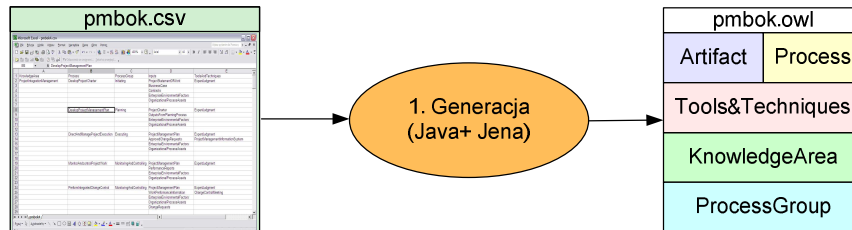


Jak powstawała ontologia PMBOK 1

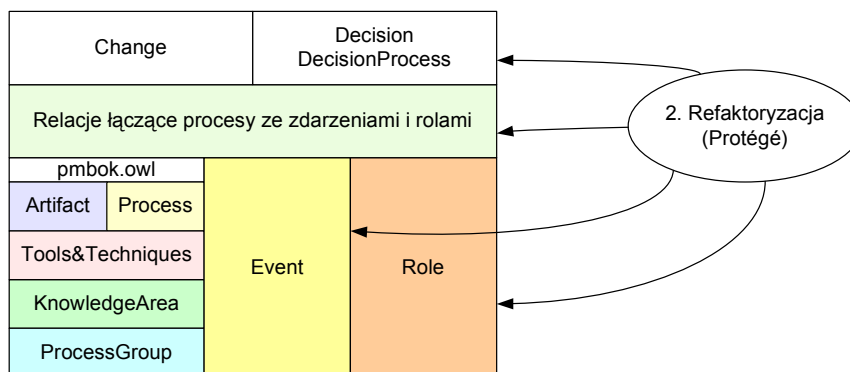
- Dwa etapy:
 - Generacja części ontologii odpowiadającej modelowi procesowemu na podstawie opisu w postaci tabeli
 - Rozszerzenie, dodanie brakujących elementów, refaktoryzacja



Jak powstawała ontologia PMBOK 2



Jak powstawała ontologia PMBOK 3





Jak powstawała ontologia Scrum?

- Ontologia metodyki Scrum powstawała w 4 kolejnych iteracjach.
 - W pierwszej iteracji na podstawie opisów literaturowych sporządzono opis tekstowy metodyki Scrum w postaci krótkich notatek definiujących znalezione pojęcia i ich relacje.
 - W drugiej pojęcia zidentyfikowane w opisie tekstowym zostały zamodelowane jako ontologia.
 - W trzeciej dokonano poprawek, usunięto redundantne pojęcia oraz uściślono pewne opisy i relacje.
 - W czwartej przeprowadzono identyfikację procesów, które jawnie nie pojawiają się w opisach Scrum.



Role



Rola - definicje

- **Funkcja.** Zbiór akcji i czynności przypisanych, oczekiwanych lub wymaganych od osoby lub grupy osób
 - X pełni funkcję **dyrektora**
 - Y pełni funkcję **kierownika projektu**
- **Funkcja.** Do czego coś jest używane?
- **Rola społeczna.** Zbiór połączonych zachowań, praw i zobowiązań. Często utożsamiana z pojęciem **aktora**. Rola wyznacza oczekiwane zachowanie w zależności od społecznego statusu i pozycji
- W **programowaniu zorientowanym na role** główną ideą jest, że człowiek myśli w kategoriach ról. Ten sam obiekt w zależności od sytuacji może odgrywać różne role:
 - Jan Kowalski jest **Studentem** uczestnicząc w zajęciach
 - Jan Kowalski jest **Uczestnikiem Przyjęcia** podczas przyjęcia
- Role są zdefiniowane przez:
 - możliwe interakcje z otoczeniem
 - wspólny zbiór własności



Role PMBOK 1

- Stakeholder \equiv Role
 - Project Performer (rola zaangażowana w realizację projektu)
 - Project Team
 - Project Development Team
 - Project Management Team
 - Project Manager
 - Functional Manager
 - Change Control Board
 - Customer (odbiorca)
 - User
 - Sponsor
 - Seller

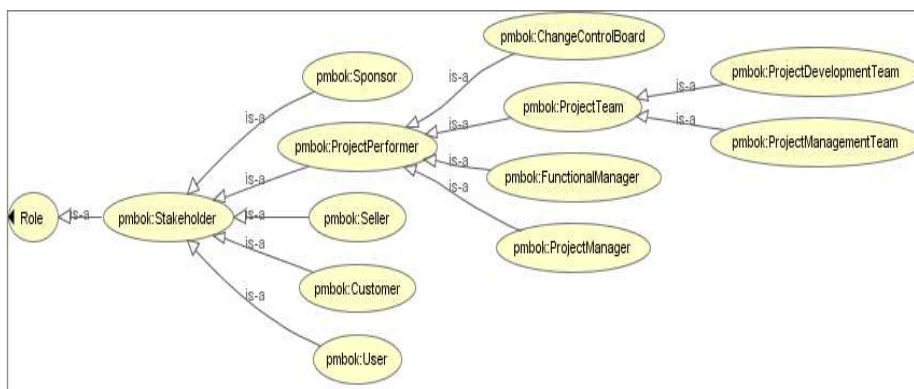


Role PMBOK 2

- **ProjectManager** odpowiada za
 - planowanie i organizację pracy
 - zarządzanie bieżącymi działaniami w ramach projektu
 - dostarczenie klientowi przedmiotów odbioru
- **FunctionalManager** odpowiada za zarządzanie dziedzinami związanymi z aspektem biznesowym:
 - zasobami ludzkimi,
 - finansami
 - kontraktami z dostawcami



Role PMBOK 3



Własności ról w PMBOK (niewidoczne na diagramie) są określone w ontologii poprzez ich uczestnictwo w procesach



Role Scrum 1

- Podstawowe role Scrum to:
 - *Stakeholder* - interesariusz,
 - *ProductOwner* - właściciel produktu
 - *ScrumMaster* - lider zespołu
 - *Team* - zespół
 - *TeamMember* - członek zespołu.



Role Scrum 2

- **Stakeholder**
 - Interesariusze, to osoby spoza grupy projektowej zainteresowane lub zależne od przebiegu prac projektowych.
 - Typowymi interesariuszami są: inwestor, kierownictwo firmy, klienci, użytkownicy i inni
- **ProductOwner** – właściciel produktu
 - Odpowiada za zarządzanie wymaganiami i nadzór nad ich realizacją w kolejności określonej przez ich wartość biznesową.
 - Jest rozliczany za sukces zespołu

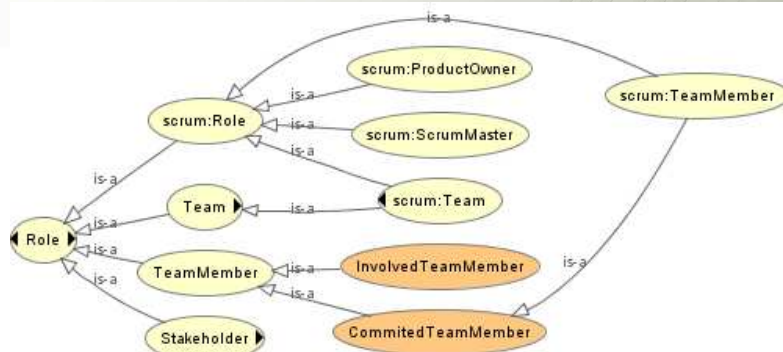


Role Scrum 3

- **ScrumMaster**
 - Dbą o zespół i jego morale
 - Dbą o postęp prac
 - Usuwa przeszkody
 - Nadzoruje zgodność praktyk projektowych z założeniami metodyki
- **Team** (zespół)
 - Składa się z członków zespołu (TeamMember)
- **TeamMember**
 - Członkowie zespołu Scrum powinni mieć umiejętności (Skills) wystarczające do rozwoju produktu
 - Powinni mieć interdyscyplinarne i pokrywające się umiejętności
 - Powinni być oddani (ang. committed) wyłącznie wykonaniu projektu, natomiast interesariusze powinni być zaangażowani w realizację projektu (ang. involved).



Role Scrum 4



- *CommittedTeamMember: TeamMember and worksOn exactly 1 Project*
- *InvolvedTeamMember: TeamMember and worksOn min 2 Project*
- *Można też próbować zdefiniować CommittedTeamMember liczbowo: TeamMember and has only ProjectEngagement that hasWorkEffortFTE exactly 1 double[>=0.95,<=1]*



Porównanie ról PMBOK i Scrum

PMBOK	Scrum
Sponsor	ProductOwner (definiowanie wymagań i ich wartości biznesowej)
ProjectManager	ProductOwner (określa priorytety wykonania) ScrumMaster (usuwanie przeszkód) Team (planowanie zadań)
FunctionalManager	ScrumMaster ProductOwner
DevelopmentTeam	Team
Stakeholder and not ProjectPerformer	Stakeholder



Artefakty




Artefakt - definicje

- **Obiekt wytworzony przez człowieka** traktowany jako pewna całość.
- **W archeologii.** Obiekt wytworzony lub ukształtowany rękami ludzkimi. Narzędzie, broń, ozdoba znaleziona na stanowisku poszukiwań archeologicznych
- **Artefakt społeczny.** Produkt jednostek lub grup osób lub przejawiane przez nie zachowanie społeczne
- **W języku UML** (Unified Modeling Language) artefakt jest to specyfikacja istniejącej fizycznie informacji, która jest używana lub wytwarzana w procesie rozwoju oprogramowania lub podczas wdrożenia lub działania systemu OMG (2008).
- Artefakt jest jednym z wielu typów materialnych produktów pośrednich wytwarzanych podczas rozwoju oprogramowania. Artefakty takie, jak przypadki użycia, diagramy klas i inne modele UML, opisy wymagań i dokumenty projektowe pozwalają opisać funkcje, architekturę i projekt oprogramowania
- **W modelowaniu.** Abstrakcyjna reprezentacja pewnego aspektu istniejącego lub budowanego systemu, komponentu lub widoku. Przykładem indywidualnych artefaktów są model graficzny, model strukturalny, dane tabelaryczne, strukturalny lub niestukturalny opis tekstowy. Indywidualne artefakty mogą być agregowane i tworzyć artefakty złożone




Artefakty PMBOK 1

- W PMBOK artefakty są wejściami i wyjściami procesów projektowych. PMBOK definiuje 75 artefaktów biorących udział w ponad 300 przepływach pomiędzy procesami.
- Definicja PMBOK nie zapewnia formalnej klasyfikacji artefaktów, przedstawiony podział został wprowadzony w trakcie refaktoryzacji wygenerowanej ontologii.
- Artefakty zostały podzielone na grupy:
- `pmbok:ArtifactGroups`
 - `pmbok:EnvironmentalArtifact`
 - `pmbok:ManagementDocument`
 - `pmbok:ProjectPlan`
 - `pmbok:ProjectBaseline`
 - `Pmbok:SubsidiaryPlan`
 - `pmbok:ProjectStatus`
 - `pmbok:ProjectDeliverables`





Artefakty PMBOK 2

- EnvironmentalArtifacts – artefakty, które przepływają pomiędzy otoczeniem i procesami projektowymi (ale nie produkty dostawy)

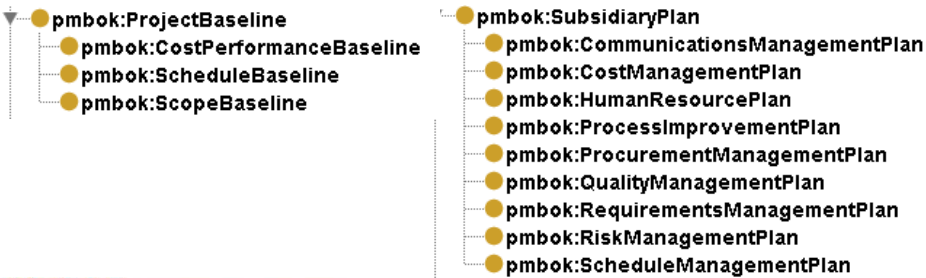


● pmbok:EnvironmentalArtifact
 ● pmbok:Contracts
 ● pmbok:ProjectStatementOfWork
 ● pmbok:BusinessCase
 ● pmbok:SellerProposals
 ● pmbok:TeamingAgreements
 ● pmbok:EnterpriseEnvironmentalFactors
 ● pmbok:EnterpriseEnvironmentalFactorsUpdates
 ● pmbok:OrganizationalProcessAssets
 ● pmbok:OrganizationalProcessAssetsUpdates


Artefakty PMBOK 3

- Plany:
 - pmbok:ManagementDocument
 - pmbok:ProjectPlan
 - pmbok:ProjectBaseline
 - pmbok:SubsidiaryPlan



● pmbok:ProjectBaseline
 ● pmbok:CostPerformanceBaseline
 ● pmbok:ScheduleBaseline
 ● pmbok:ScopeBaseline

● pmbok:SubsidiaryPlan
 ● pmbok:CommunicationsManagementPlan
 ● pmbok:CostManagementPlan
 ● pmbok:HumanResourcePlan
 ● pmbok:ProcessImprovementPlan
 ● pmbok:ProcurementManagementPlan
 ● pmbok:QualityManagementPlan
 ● pmbok:RequirementsManagementPlan
 ● pmbok:RiskManagementPlan
 ● pmbok:ScheduleManagementPlan





Artefakty PMBOK 4

- Pozostałe plany

- pmbok:ActivityAttributes
- pmbok:ActivityCostEstimates
- pmbok:ActivityDurationEstimates
- pmbok:ActivityList
- pmbok:ActivityResourceRequirements
- pmbok:BasisOfEstimates
- pmbok:MakeOrBuyDecisions
- pmbok:MilestoneList
- pmbok:OutputsFromPlanningProcess
- pmbok:ProjectManagementPlan
- pmbok:ProjectManagementPlanUpdates
- pmbok:ProjectSchedule
- pmbok:ProjectScheduleNetworkDiagram
- pmbok:ProjectStaffAssigned
- pmbok:ProjectStaffAssignments
- pmbok:ResourceBreakdownStructure
- pmbok:ResourceCalendars
- pmbok:Risk-relatedContractDecisions
- pmbok:RiskRelatedContractDecisions
- pmbok:ScheduleData
- pmbok:StakeholderManagementStrategy
- pmbok:WBS
- pmbok:WBSDictionary




Artefakty PMBOK 5

- Grupa



ManagementDocument
ProjectStatus

- pmbok:ProjectStatus
- pmbok:ApprovedChangeRequests
- pmbok:BudgetForecasts
- pmbok:ChangeLog
- pmbok:IssueActionItemLog
- pmbok:PerformanceReports
- pmbok:ProcurementsContractAward
- pmbok:QualityControlMeasurements
- pmbok:RiskRegister
- pmbok:RiskRegisterUpdates
- pmbok:SelectedSellers
- pmbok:TeamPerformanceAssessments
- pmbok:WorkPerformanceInformation
- pmbok:WorkPerformanceMeasurements



Artefakty PMBOK 6

- Pozostałe dokumenty zarządzania (nie sklasyfikowane jako plany i dokumenty o statusie projektu)
 - pmbok:ChangeRequests
 - pmbok:ChangeRequestsStatusUpdate
 - pmbok:ClosedProcurements
 - pmbok:ProcurementDocuments
 - pmbok:ProcurementStatementsOfWork
 - pmbok:ProjectCharter
 - pmbok:ProjectDocuments
 - pmbok:ProjectDocumentsUpdates
 - pmbok:ProjectFundingRequirement
 - pmbok:ProjectScopeStatement
 - pmbok:QualifiedSellerList
 - pmbok:QualityChecklists
 - pmbok:QualityMetrics
 - pmbok:RequirementsDocumentation
 - pmbok:RequirementsTraceabilityMatrix
 - pmbok:SourceSelectionCriteria
 - pmbok:StakeholderRegister
 - pmbok:ValidatedChanges






Artefakty PMBOK 7

- ProjectDeliverables (produkty dostawy)
 - pmbok:ProjectDeliverables
 - pmbok:AcceptedDeliverables
 - pmbok:Deliverables
 - pmbok:FinalProductOrServiceOrResultTransition
 - pmbok:ValidatedDeliverables
- Możliwa (konieczna) refaktoryzacja

ValidatedDeliverables ≡ Deliverables and hasStatus value Validated

AcceptedDeliverables ≡ Deliverables and hasStatus value Accepted






Artefakty Scrum 1

- `scrum:Artifact`
 - `scrum:Budget`
 - `scrum:BurndownChart`
 - `scrum:ImpedimentBacklog`
 - `scrum:ImplementedFeature`
 - `scrum:InitialOverallRequirements`
 - `scrum:LessonsLearned`
 - `scrum:MeetingIssues`
 - `scrum:ProductBacklog`
 - `scrum:ProductBacklogItem`
 - `scrum:ProductBaseline`
 - `scrum:ProductRoadMap`
 - `scrum:ProductStatus`
 - ▼ `scrum:ProductSubrelease`
 - `scrum:ProductRelease`
 - `scrum:ProductVision`
 - `scrum:ROIObjectives`
 - `scrum:ReleaseGoal`
 - `scrum:ReleasePlan`
 - `scrum:ReleaseStrategy`
 - `scrum:RequestedProductChange`
 - `scrum:SprintBacklog`
 - `scrum:SprintGoals`
 - `scrum:TaskDescription`
 - `scrum:TeamCapabilities`
 - `scrum:WorkDone`

- Charakterystyczne artefakty:
 - *ProductBacklog* - rejestr produktowy
 - *SprintBacklog* - rejestr zadaniowy
 - ImpedimentsBacklog
 - BurndownChart - wykres wypalania
 - Dla Sprintu
 - Dla produktu




Artefakty Scrum 2

- ProductBacklog
 - stwierdzenie – „rejestr produktowy zawiera wymagania” uściślone przez relacje pomiędzy klasami:
ProductBacklog has some ProductBacklogItem
 - *ProductBacklogItem* jest określony przez szereg własności

Superclasses +

● <code>scrum:Artifact</code>
● hasProperty some <code>scrum:AdjustedEstimate</code>
● hasProperty some <code>scrum:ComplexityFactor</code>
● hasProperty some <code>scrum:EstimatedBusinessValue</code>
● hasProperty some <code>scrum:InitialEstimate</code>
● refersTo some <code>ProductRequirement</code>
● isManagedBy exactly 1 <code>scrum:ProductOwner</code>
● hasName exactly 1 string



Poland Chapter



Artefakty Scrum 3

- **SprintBacklog**

- stwierdzenie – „rejestr zadaniowy zawiera zadania” uściślone przez relacje pomiędzy klasami:

SprintBacklog has some TaskDescription

- **TaskDescription** jest określony przez własności

Superclasses +	
●	scrum:Artifact
●	originator some TeamMember
●	responsible exactly 1 TeamMember
●	status exactly 1 {NotStarted, InProgress, Completed}
●	estimatedRemainingTime exactly 1 (double[>= "0"^^integer])
●	hasName exactly 1 string



Artefakty Scrum 4

- **ImpedimentBacklog** zawiera elementy typu ImpedimentBacklogItem

Description: scrum:ImpedimentBacklogItem	
Equivalent classes +	
Superclasses +	
●	scrum:Artifact
●	originator some scrum:TeamMember
●	responsible some (scrum:TeamMember or scrum:ScrumMaster)
●	status some {New,Assigned, Resolved}
●	hasName exactly 1 string
●	hasPriority exactly 1 integer



Odzworowanie artefaktów PMBOK i Scrum

PMBOK	Scrum
WBS Requirement document MilestoneList	ProductBacklog
ActivityList MilestoneList	SprintBacklog
WorkPerformance Measurements WorkPerformanceInformation (Earned Value Diagrams)	BurndownChart



Zdarzenia



Zdarzenia – definicja 1 [Wikipedia]

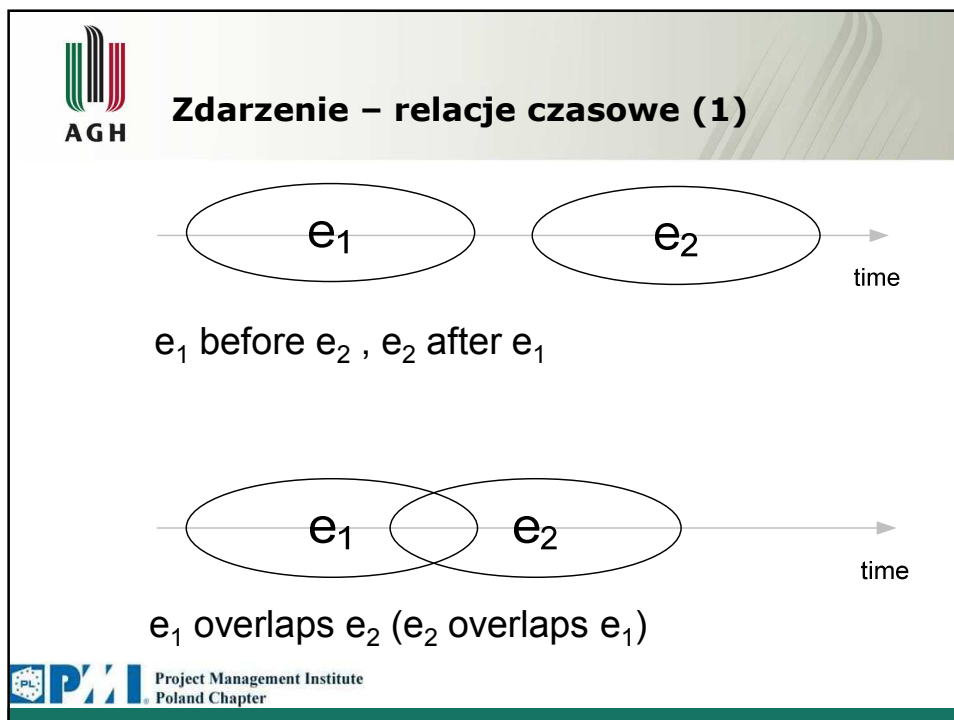
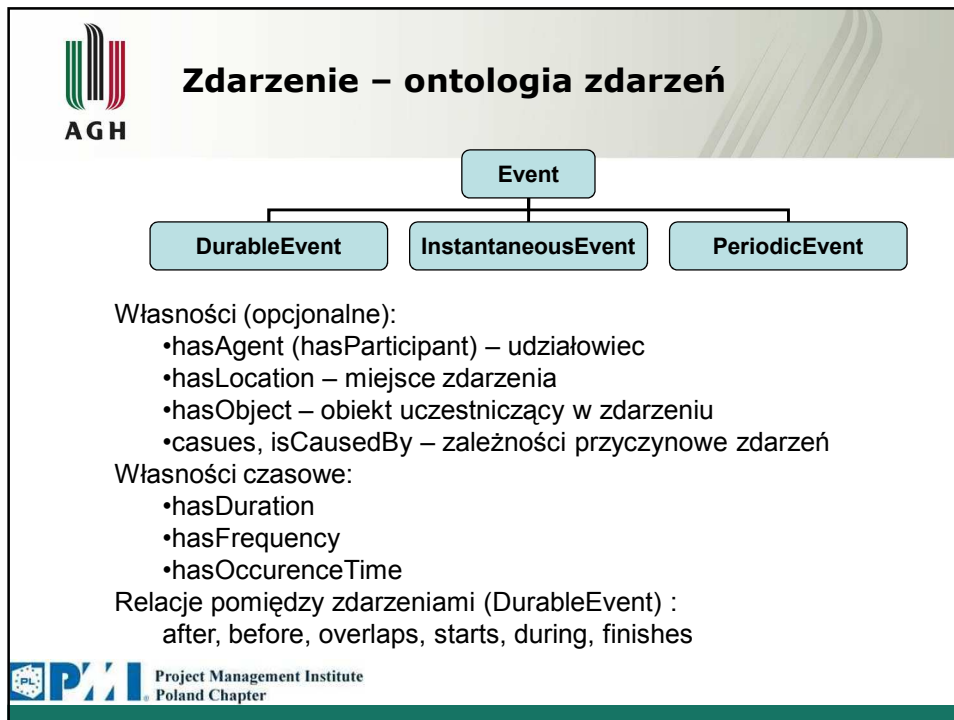
- Fenomen, obserwowalne wystąpienie lub nadzwyczajne wystąpienie (powódź, trzęsienie ziemi)
- Rodzaj zebrania:
 - Ceremonia (ślub)
 - Zawody
 - Zjazd, spotkanie
 - Przedstawienie, festiwal, sytuacja uznawana jako przejaw działalności artystycznej
 - Wydarzenie medialne (zdarzenie przyciągające uwagę mediów)
 - Przyjęcie, impreza
 - Wydarzenie sportowe
- W informatyce: komunikat sygnalizujący, że coś się zdarzyło (naciśnięcie klawisza lub przycisku myszy)

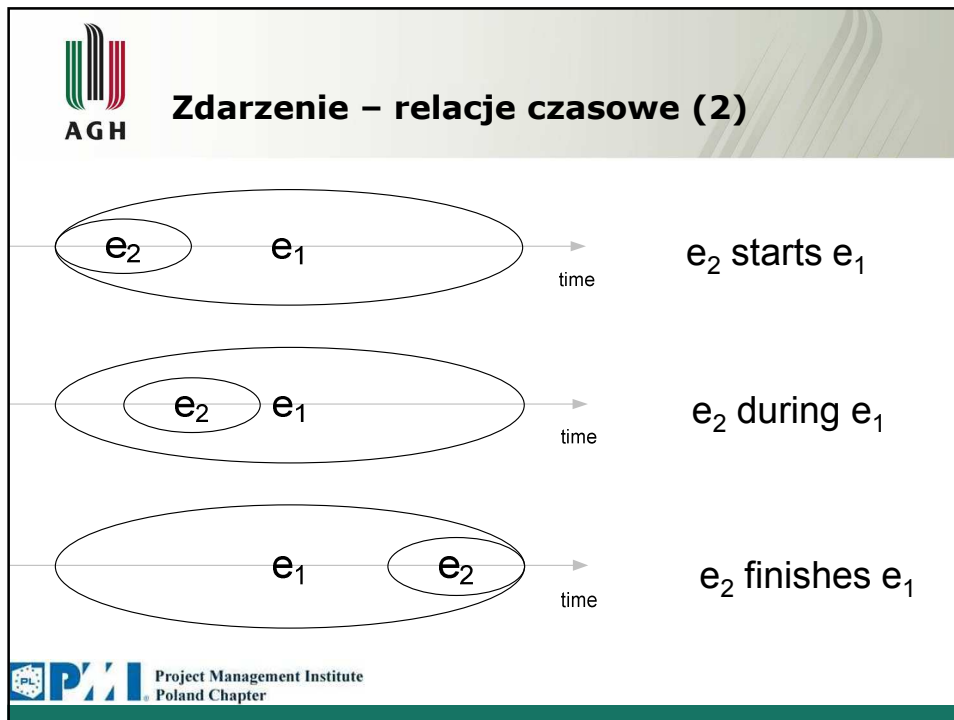



Zdarzenia – definicja 2 [Wikipedia]

- **W fizyce** – sytuacja występująca w pewnym miejscu i czasie (obecność cząstki w czasoprzestrzeni)
- **W filozofii** - obiekt w czasie lub instancjacja pewnej cechy obiektu w czasie
- **Jaegwon Kim**
 - Zdarzenie jest strukturą $[x,P,t]$, gdzie
 - x – obiekt lub zbiór obiektów
 - P – zbiór własności
 - t – moment czasowy lub interwał


Zdarzenie występuje, jeżeli obiekt x przejawia własności P w momencie t





 **Zdarzenia i procesy**

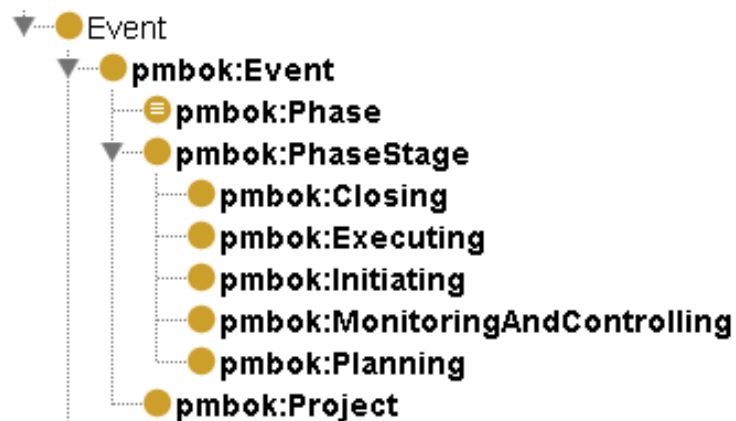
- Konstruując ontologie Scrum i PMBOK rozdzielono zdarzenia i procesy:
 - **Zdarzenia** są zdefiniowane poprzez **relacje czasowe** (czas wystąpienia, trwania, itd.) oraz **własności** (uczestników, miejsce, obiekt)
 - **Procesy** transformują wejścia w wyjścia i są uruchamiane podczas zdarzeń lub w wyniku pojawienia się zdarzenia
- Dygresja
 - **Zadanie** (task) to proces lub jego część (aktywność) z przypisanymi
 - parametrami czasowymi (czas rozpoczęcia, zakończenia, oszacowanie czasu trwania)
 - Wykonawcami
 - Nie mówimy, *Event missed deadline* lub *Process missed deadline*, ale: *Task missed deadline...*

 Project Management Institute
Poland Chapter



Zdarzenia PMBOK 1

- Podstawowe zdarzenia PMBOK



Zdarzenia PMBOK 2

- Projekt (Project)

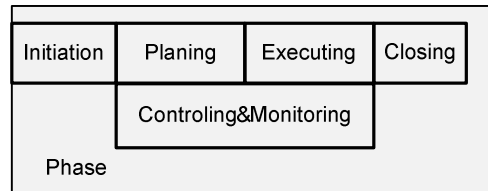
● pmbok:Event
 ● DurableEvent
 ● consistsOf **min** 1 pmbok:Phase

- Faza (Phase)

● pmbok:Event
 and DurableEvent
 and consistsOf **exactly** 1 pmbok:Closing
 and consistsOf **exactly** 1 pmbok:Executing
 and consistsOf **exactly** 1 pmbok:Initiating
 and consistsOf **exactly** 1 pmbok:MonitoringAndControlling
 and consistsOf **exactly** 1 pmbok:Planning



Zdarzenia PMBOK 3 Zależności czasowe wewnątrz fazy

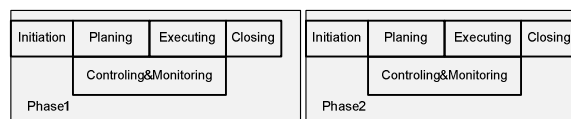


Initiating starts some Phase
Initiating before some Planning
Planning overlaps some Executing
Closing finishes some Phase
 itd...

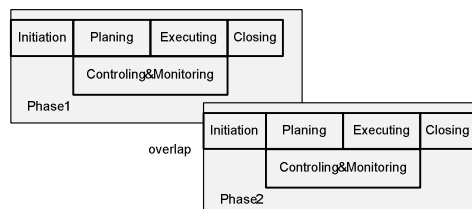


Zdarzenia PMBOK 4 Relacje pomiędzy fazami 1

- Sekwencyjna



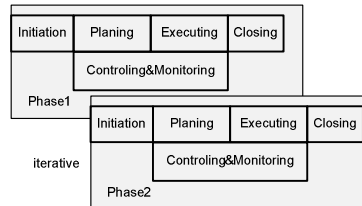
- Nakładanie się (overlaps)



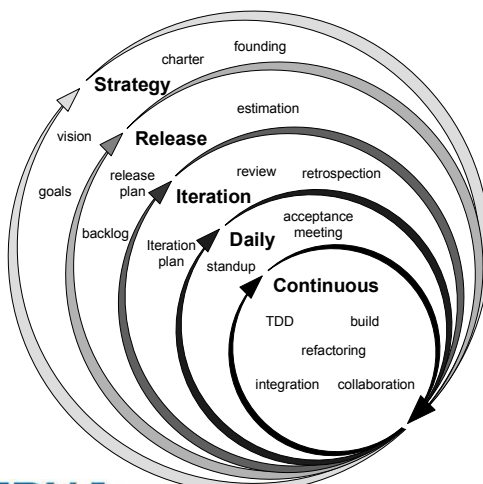


Zdarzenia PMBOK 5 Relacje pomiędzy fazami 2

- Iteracyjna (podczas etapu wykonania fazy odbywa się planowanie następnej fazy)



Zdarzenia Scrum 1



- Wykres cebulkowy wg. Mike Cohna (zmodyfikowany przez VersionOne Inc.) definiuje hierarchię zawierania się horyzontów planowania

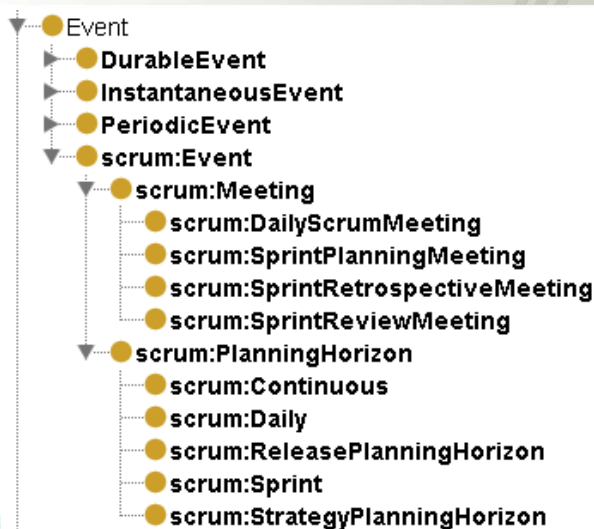


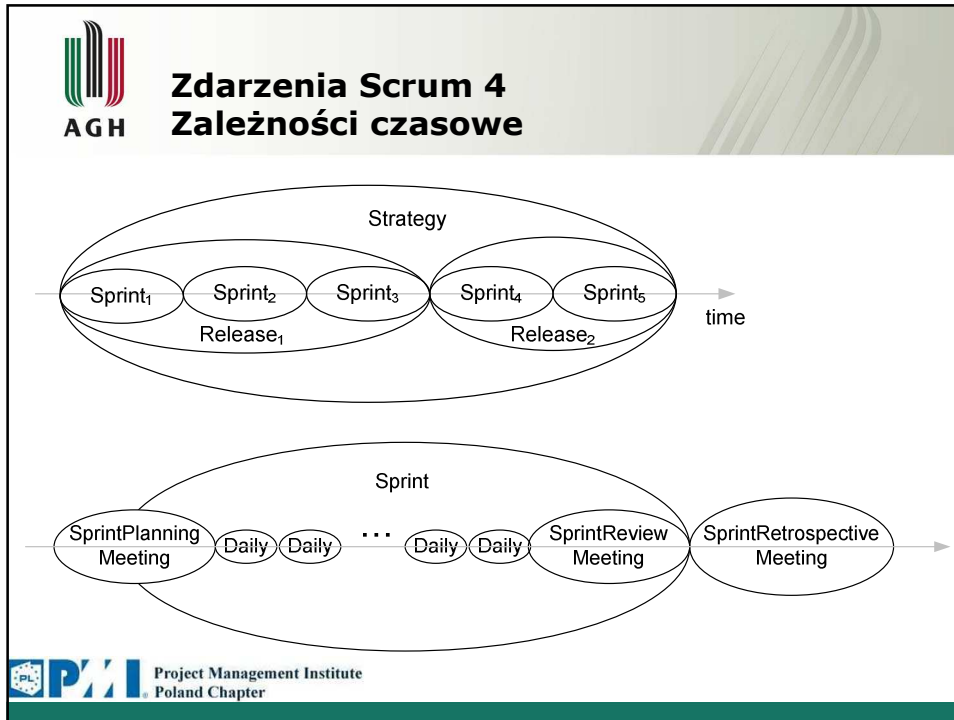
Zdarzenia Scrum 2


- Zebrania pojawiające się w opisach praktyk Scrum:
 - *SprintPlanningMeeting* - planowanie zakresu iteracji
 - *DailyScrumMeeting* - codzienne spotkanie przeznaczone na omówienie postępu prac i planowanie
 - *SprintReviewMeeting* - przegląd rezultatów osiągniętych w trakcie iteracji
 - *SprintRetrospectiveMeeting* - sformułowanie wniosków z przebiegu iteracji i zaleceń na przyszłość





Zdarzenia Scrum 3 Klasyfikacja




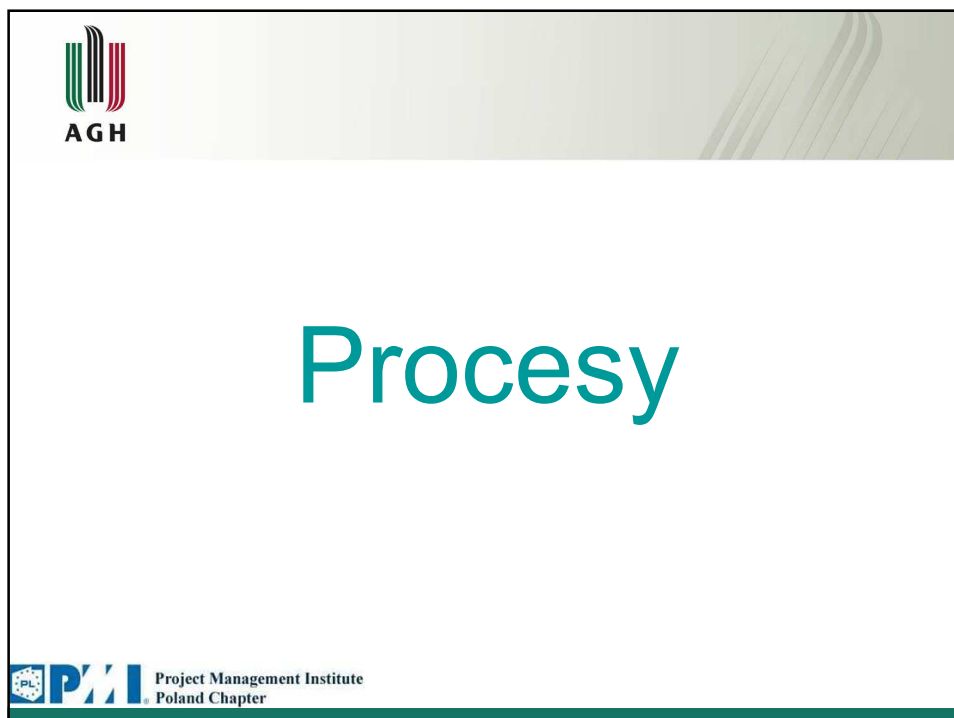
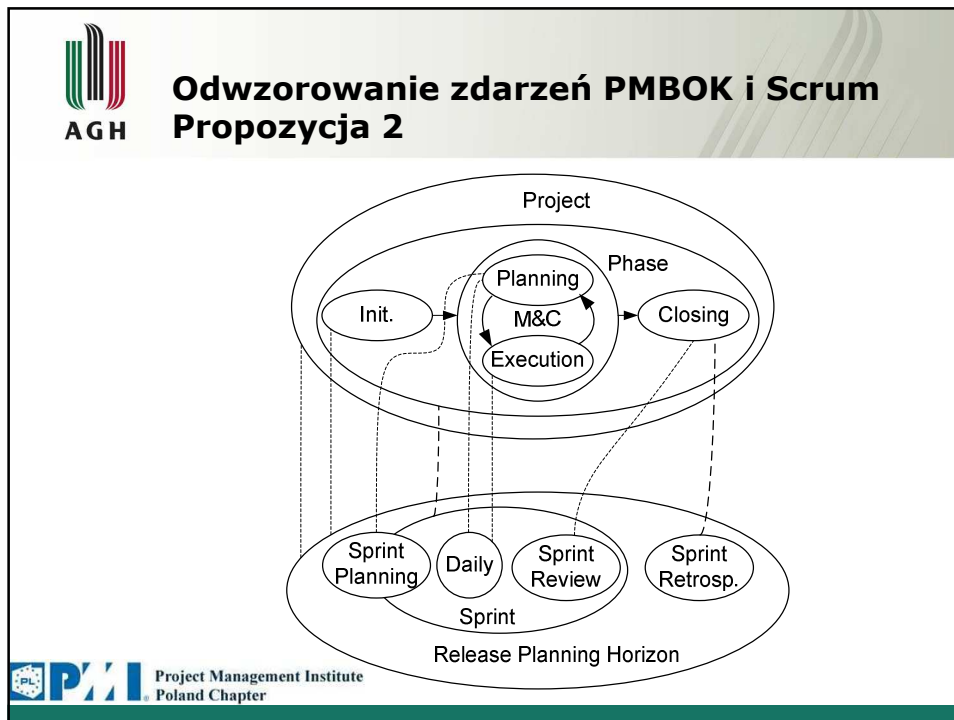


 Odwzorowanie zdarzeń PMBOK i Scrum Propozycja 1 Phase=ReleasePlanningHorizon	
PMBOK	Scrum
pmbok:Project	scrum:StartegyPlanningHorizon
pmbok:Phase	scrum:ReleasePlanningHorizon
pmbok:Initiating	scrum:ReleasePlanningHorizon
pmbok:Planning	Wielokrotne: scrum:SprintPlanningMeeting scrum:Sprint, scrum:Daily
pmbok:Executing	scrum:Sprint, scrum:Daily
pmbok:MonitorngAndControlling	scrum:Daily, scrum:SprintReviewMeeting, scrum:SprintRetrospectiveMeeting
pmbok:Closing	scrum:SprintReviewMeeting (związane z dostarczeniem wydania)

 Project Management Institute
 Poland Chapter

 Odwzorowanie zdarzeń PMBOK i Scrum Propozycja 2 Phase=Sprint	
PMBOK	Scrum
Brak odwzorowania	scrum:StartegyPlanningHorizon
pmbok:Project	scrum:ReleasePlanningHorizon
pmbok:Phase	scrum:Sprint
pmbok:Initiating	scrum:ReleasePlanningHorizon Równolegle w trakcie trwania Sprintu
pmbok:Planning	scrum:SprintPlanningMeeting Scrum:Daily
pmbok:Executing	scrum:Daily
Pmbok:MonitorngAndControlling	scrum:Daily scrum:SprintRetrospectiveMeeting
pmbok:Closing	scrum:ReviewMeeting

 Project Management Institute
 Poland Chapter





Proces definicja

- Ciąg zachodzących po sobie powiązanych procedur (zmian, stanów), które podczas kolejnych etapów konsumują zasoby (ludzkie, czas, energia, sprzęt, pieniądze), aby przekonwertować wejścia w wyjścia. Wyjścia służą jako wejścia kolejnych etapów, aż osiągnięty zostanie oczekiwany cel lub rezultat działania procesu.



Proces – model ontologiczny

Description: Process

Equivalent classes +

Superclasses +

- (hasTrigger some ProcessTrigger) or (enabledDuring some DurableEvent) or (disabledDuring some DurableEvent)
- hasInput some Artifact
- hasOutput some Artifact
- isExecutedBy some Role
- hasToolsAndTechniques min 0 ToolsAndTechniques



PMBOK procesy 1

- PMBOK definiuje 42 procesy podając ich:
 - Wejściowe artefakty (relacja *hasInput*)
 - Wyjściowe artefakty (relacja *hasOutput*)
 - Narzędzia i techniki (*hasToolsAndTechniques*). Podano w sumie 129 narzędzi i technik.
- Procesom nie są jednoznacznie przypisane role odpowiadające za ich wykonanie (*isExecutedBy*)
- W PMBOK sklasyfikowano procesy na dwa sposoby:
 - Poprzez przypisanie do grupy procesów (ProcessGroup)
 - Poprzez przypisanie do obszaru wiedzy (KnowledgeArea)



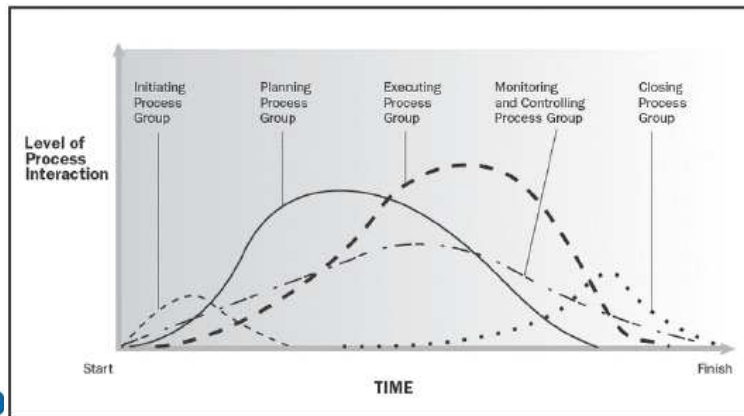
PMBOK procesy 2 Grupy procesów

- PMBOK definiuje następujące grupy procesów:
 - InitiatingProcessGroup
 - PlanningProcesGroup
 - ExecutingProcessGroup
 - MonitoringAndControllingProcessGroup
 - ClosingProcessGroup
- Przynależność procesu do grupy zamodelowano jako relację:
Process belongsToProcessGroup some ProcessGroup
- Równocześnie zdefiniowano relację odwrotną łączącą grupy procesów z procesami:
ProcessGroup containsProcess some Process



PMBOK procesy 3 Grupy procesów

- Podział procesów na grupy był źródłem identyfikacji zdarzeń wewnątrz fazy; proces należący do danej grupy jest wykonywany podczas odpowiadającego jej zdarzenia



PMBOK procesy 3 Obszary wiedzy

- Drugą klasyfikacją jest przypisanie procesów do obszarów wiedzy
- Przynależność procesu do obszaru wiedzy zamodelowano jako relację:
Process belongsToKnowledgeArea some KnowledgeArea
- Równocześnie zdefiniowano relację odwrotną łączącą grupy procesów z procesami:
KnowledgeArea coversProcess some Process



PMBOK procesy 4 Obszary wiedzy



PMBOK procesy 5 Przykład definicji procesu pmbok:CollectRequirements

● pmbok:Process
● belongsToKnowledgeArea some pmbok:ProjectScopeManagement
● belongsToProcessGroup some pmbok:PlanningProcessGroup
● hasInput some pmbok:ProjectCharter
● hasInput some pmbok:StakeholderRegister
● hasOutput some pmbok:RequirementsDocumentation
● hasOutput some pmbok:RequirementsManagementPlan
● hasOutput some pmbok:RequirementsTraceabilityMatrix
● hasToolsAndTechniques some pmbok:FacilitatedWorksHops
● hasToolsAndTechniques some pmbok:FocusGroups
● hasToolsAndTechniques some pmbok:GroupCreativityTechniques
● hasToolsAndTechniques some pmbok:Interviews
● hasToolsAndTechniques some pmbok:Observations
● hasToolsAndTechniques some pmbok:Prototypes
● hasToolsAndTechniques some pmbok:QuestionnairesAndsurveys



PMBOK procesy 6 Przykład definicji procesu pmbok:CollectRequirements

- Elementy wymagające uściślenia – ToolsAndTechniques
- Przykład (z dokumentacji):
GroupCreativityTechniques
 - Brainstorming
 - NominalGroupTechnique
 - DelphiTechnique
 - IdeaMindMapping
 - AffinityDiagram
- Obecnie brak pełnych klasyfikacji



Scrum procesy 1

- Procesy fazy opracowywania strategii
 - 1.1. Vision Creation
 - 1.2. Product RoadMap Creation
- Procesy fazy planowania wydania
 - 2.1. Release Planning
 - 2.2. Initial Backlog Creation
 - 2.3. Backlog Updating
- Procesy fazy Sprintu (iteracji)
 - 3.1. Sprint Scope Planning
 - 3.2. Initial Sprint Planning
 - 3.3. Sprint Review
 - 3.4. Sprint Retrospective
- Procesy codzienne
 - 4.1. Sprint Tasks Control
 - 4.2. Sprint Tasks Updating
 - 4.3. Impediments Reporting
 - 4.4. Feature Development



Scrum procesy 2 Faza opracowania strategii 1.1. Vision Creation

Description: scrum:VisionCreation

Superclasses +

- **scrum:Process**
- hasInput **some** scrum:InitialOverallRequirements
- hasInput **some** scrum:ROIObjectives
- hasOutput **some** scrum:ProductVision
- isExecutedBy **some** scrum:FoundingStakeholder
- enabledDuring **exactly** 1 (scrum:StrategyPlanningHorizon
and not scrum:ReleasePlanningHorizon)



Scrum procesy 3 Faza opracowania strategii 1.2. Product RoadMap Creation

Description: scrum:ProductRoadMapCreation

Superclasses +

- **scrum:Process**
- hasInput **some** scrum:Budget
- hasInput **some** scrum:ProductVision
- hasOutput **some** scrum:ProductRoadMap
- isExecutedBy **some** scrum:ProductOwner
- enabledDuring **exactly** 1 (scrum:StrategyPlanningHorizon
and not scrum:ReleasePlanningHorizon)



Scrum procesy 4 Faza planowania wydania 2.1. Release Planning

Description: `scrum:ReleasePlanning`

Superclasses +

- `scrum:Process`
- enabledDuring **some** (`scrum:ReleasePlanningHorizon` and not `scrum:Sprint`)
- hasInput **some** `scrum:ProductRoadMap`
- hasOutput **some** `scrum:ReleaseGoal`
- hasOutput **some** `scrum:ReleasePlan`
- hasOutput **some** `scrum:ReleaseStrategy`
- isExecutedBy **some** `scrum:ProductOwner`
- isExecutedBy **some** `scrum:Stakeholder`



Scrum procesy 5 Faza planowania wydania 2.2. Initial Backlog Creation

Description: `scrum:InitialBacklogCreation`

Superclasses +

- `scrum:Process`
- enabledDuring **some** (`scrum:ReleasePlanningHorizon` and not `scrum:Sprint`)
- hasInput **some** `scrum:ReleasePlan`
- hasOutput **some** (`scrum:ProductBacklog` and isOrderedBy **some** `scrum:Priority`)
- isExecutedBy **some** `scrum:ProductOwner`
- isExecutedBy **some** `scrum:Stakeholder`



Scrum procesy 6 Faza planowania wydania 2.3. Backlog Updating

Description: `scrum:BacklogUpdating`

Superclasses +

- `scrum:Process`
- enabledDuring **some** (`scrum:ReleasePlanningHorizon` and not `scrum:SprintPlanningMeeting`)
- hasInput **some** `scrum:ProductBacklog`
- hasInput **some** `scrum:RequestedProductChange`
- hasOutput **some** (`scrum:ProductBacklog` and isOrderedBy **some** `scrum:Priority`)
- isExecutedBy **some** `scrum:ProductOwner`
- isExecutedBy **some** `scrum:Stakeholder`




Scrum procesy 7 Faza Sprintu 3.1. Sprint Scope Planning

Description: `scrum:SprintScopePlanning`

Equivalent classes +

Superclasses +

- `scrum:Process`
- enabledDuring **some** `scrum:SprintPlanningMeeting`
- hasInput **some** `scrum:ProductBacklog`
- hasInput **some** `scrum:ProductBaseline`
- hasOutput **some** `scrum:ProductBacklog`
- hasTrigger **some** `scrum:SprintPlanningMeeting`
- isExecutedBy **some** `scrum:ProductOwner`


 **Scrum procesy 8**
Faza Sprintu
3.2. Initial Sprint Planning


Description: `scrum:InitialSprintPlanning`

Equivalent classes +

Superclasses +

- `scrum:Process`
- enabledDuring **some** `scrum:SprintPlanningMeeting`
- hasInput **some** `scrum:ProductBacklog`
- hasInput **some** `scrum:ProductBaseline`
- hasInput **some** `scrum:ProductVision`
- hasInput **some** `scrum:TeamCapabilities`
- hasOutput **some** `scrum:ImpedimentBacklog`
- hasOutput **some** `scrum:SprintBacklog`
- hasOutput **some** `scrum:SprintGoals`
- isExecutedBy **some** `scrum:ProductOwner`
- isExecutedBy **some** `scrum:Team`

 Project Management Institute
Poland Chapter


 **Scrum procesy 9**
Faza Sprintu
3.3. Sprint Review

Description: `scrum:SprintReview`

Equivalent classes +

Superclasses +

- `scrum:Process`
- enabledDuring **some** `scrum:SprintReviewMeeting`
- hasInput **some** `scrum:ProductBaseline`
- hasOutput **some** `scrum:WorkDone`
- isExecutedBy **some** `scrum:ProductOwner`
- isExecutedBy **some** `scrum:Stakeholder`
- isExecutedBy **some** `scrum:Team`

 Project Management Institute
Poland Chapter



Scrum procesy 10 Faza Sprintu 3.4. Sprint Retrospective

Description: `scrum:SprintRetrospective`

Equivalent classes

Superclasses

- `scrum:Process`
- `enabledDuring some scrum:SprintRetrospectiveMeeting`
- `hasInput some scrum:ProductStatus`
- `hasInput some scrum:WorkDone`
- `hasOutput some scrum:LessonsLearned`
- `isExecutedBy some scrum:ScrumMaster`
- `isExecutedBy some scrum:Team`



Scrum procesy 11 Faza Daily 4.1. Sprint Tasks Control

Description: `scrum:SprintTasksControl`

Equivalent classes

Superclasses

- `scrum:Process`
- `enabledDuring some scrum:DailyScrumMeeting`
- `hasInput some scrum:SprintBacklog`
- `hasOutput some scrum:BurndownChart`
- `isExecutedBy some scrum:ProductOwner`




Scrum procesy 12 Faza Daily 4.2. Sprint Tasks Updating

Description: <code>scrum:SprintTasksUpdating</code>
Equivalent classes +
Superclasses +
● <code>scrum:Process</code>
● enabledDuring some <code>scrum:DailyScrumMeeting</code>
● hasInput some <code>scrum:SprintBacklog</code>
● hasOutput some <code>scrum:SprintBacklog</code>
● isExecutedBy some <code>scrum:Team</code>



Scrum procesy 13 Faza Daily 4.3. Impediments Reporting

Description: <code>scrum:ImpedimentsReporting</code>
Equivalent classes +
Superclasses +
● <code>scrum:Process</code>
● enabledDuring some <code>scrum:DailyScrumMeeting</code>
● hasInput some <code>scrum:MeetingIssues</code>
● hasOutput some <code>scrum:ImpedimentBacklog</code>
● isExecutedBy some <code>scrum:Team</code>





Scrum procesy 14 Faza Daily 4.4. Feature Development

Description: scrum:FeatureDevelopment

Equivalent classes +


Superclasses +

- **scrum:Process**
- enabledDuring **some** scrum:Daily
- hasInput **some** scrum:SprintBacklog
- hasOutput **some** scrum:ImplementedFeature
- hasOutput **some** scrum:ProductBacklog
- hasOutput **some** scrum:SprintBacklog
- isExecutedBy **some** scrum:Team

Odzworowanie procesów PMBOK i Scrum?

- Jest planowane jako następny etap prac
- Powinno uwzględniać odzworowanie:
 - Artefaktów (wejść i wyjść procesów)
 - Zdarzeń, podczas których wykonywane są procesy
 - Ról biorących udział w ich wykonaniu
- Planowane jest zastosowanie strukturalnych technik odzworowania klas (pojęć) w ontologiach.





Zastosowania



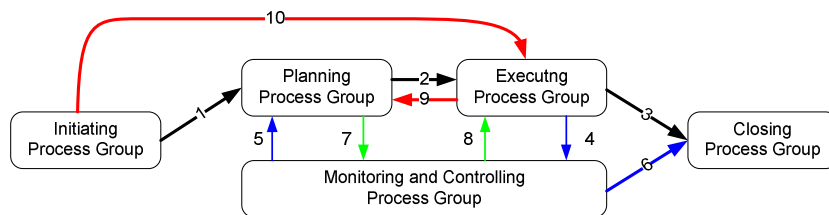
Zastosowania

- Obecne możliwe zastosowania
 - Eksploracja modeli ontologicznych
 - nawigacja w Protégé
 - przeglądanie wygenerowanych stron HTML
 - zapytania
 - Wykrywanie błędów i niespójności
- Przyszłe zastosowania
 - Odwzorowanie ontologii
 - Integracja
 - Skalowanie i dekompozycja

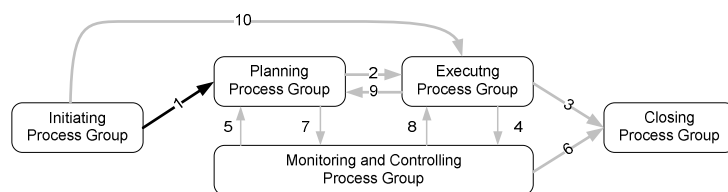


Zastosowania Eksploracja modelu PMBOK

- Poszukiwanie artefaktów przepływających pomiędzy grupami procesów



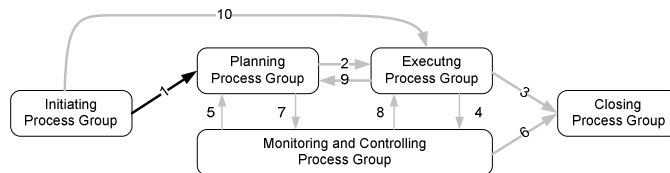
Zastosowania Przykład eksploracji modelu PMBOK 1



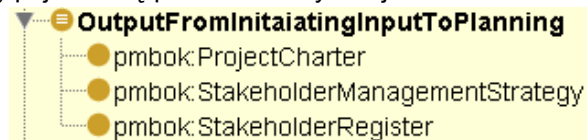
- Definiujemy sztuczną klasę (kwerendę) **OutputFromInitiatingInputToPlanning** jako:
pmbok:Artifact
and isInputTo some (pmbok:Process
and belongsToProcessGroup some pmbok:PlanningProcessGroup)
and isOutputFrom some (pmbok:Process
and belongsToProcessGroup some pmbok:InitiatingProcessGroup)



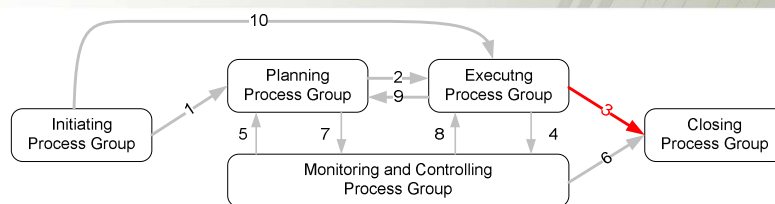
Zastosowania Przykład eksploracji modelu PMBOK 2



Uruchamiamy maszynę wnioskującą, np.. Fact++ w Protégé. W zbudowanej na podstawie wnioskowania hierarchii klas (inferred class hierarchy) pojawia się poniższa klasyfikacja



Zastosowania Przykład eksploracji modelu PMBOK 3



- Czy przepływ (3) jest reprezentowany przez jakieś artefakty?
- Definiujemy kwerendę `OutputFromExecutingInputToClosing`
`pmbok:Artifact`
`and isInputTo some (pmbok:Process`
`and belongsToProcessGroup some pmbok:ClosingProcessGroup)`
`and isOutputFrom some (pmbok:Process`
`and belongsToProcessGroup some pmbok:ExecutingProcessGroup)`
- Odpowiedź: **nie, brak artefaktów bezpośrednio przekazywanych z grupy Executing do Closing**

AGH Zastosowania
Przykład eksploracji modelu PMBOK 4

```

    graph LR
      Initiating -- 1 --> Planning
      Planning -- 2 --> Executing
      Executing -- 3 --> Closing
      Monitoring[Monitoring and Controlling] -- 4 --> Executing
      Monitoring -- 5 --> Planning
      Monitoring -- 6 --> Closing
      Monitoring -- 7 --> Planning
      Monitoring -- 8 --> Executing
      Executing -- 9 --> Planning
      Initiating -- 10 --> Closing
  
```

- Czy przepływ (9) jest reprezentowany przez artefakty?

OutputFromExecutionInputToPlanning
pmbok:ResourceCalendars

PMI Project Management Institute
Poland Chapter


AGH Zastosowania
Przykład eksploracji modelu PMBOK 5

Szczegółowa analiza ResourceCalendars

Description: pmbok:ResourceCalendars	
Equivalent classes	+
Superclasses	+
● pmbok:ProjectPlan	
● pmbok:Artifact	
● isInputTo some pmbok:DetermineBudget	
● isInputTo some pmbok:DevelopProjectTeam	
● isInputTo some pmbok:DevelopSchedule	
● isInputTo some pmbok:EstimateActivityDurations	
● isInputTo some pmbok:EstimateActivityResources	
● isOutputFrom some pmbok:AcquireProjectTeam	
● isOutputFrom some pmbok:ConductProcurements	

- Jest wyjściem **AcquireProjectTeam** (pewnego procesu wykonania)
- Jest wejściem **DevelopSchedule** (pewnego procesu planowania)

PMI Project Management Institute
Poland Chapter




Zastosowania

Przykład eksploracji modelu PMBOK 6


Błędy w PMBOK?

- Artefakty reprezentujące przepływ (5)?
- **Błąd:** Lista artefaktów pusta, oczekiwana ApprovedChangeRequest?

Kwerenda	Definicja	Wynik
ProcessProducing ApprovedChangeRequests	pmbok:Process and hasOutput some pmbok: ApprovedChangeRequests	Brak takich procesów
ProcessProducing ChangeRequestsStatusUpdates	pmbok:Process and hasOutput some pmbok: ChangeRequestsStatusUpdate	pmbok: PerformIntegratedChangeControl



Project Management Institute
Poland Chapter



Zastosowania

Błędy w PMBOK?

9.4 Manage Project Team

- 1 Inputs
 - 1 Project staff assignments
 - 2 Project management plan
 - 3 Team performance assessments
 - 4 Performance reports
 - 5 Organizational process assets
- 2 Tools & Techniques
 - 1 Observation and conversation
 - 2 Project performance appraisals
 - 3 Conflict management
 - 4 Issue log
 - 5 Interpersonal skills
- 3 Outputs
 - 1 Enterprise environmental factors updates
 - 2 Organizational process assets updates
 - 3 Change requests
 - 4 Project management plan updates

10.4 Manage Stakeholder Expectations


- 1 Inputs
 - 1 Stakeholder register
 - 2 Stakeholder management strategy
 - 3 Project management plan
 - 4 Issue log
 - 5 Change requests
 - 6 Organizational process assets
- 2 Tools & Techniques
 - 1 Communication methods

Description: pmbok:IssueLog


Equivalent classes +

Superclasses +

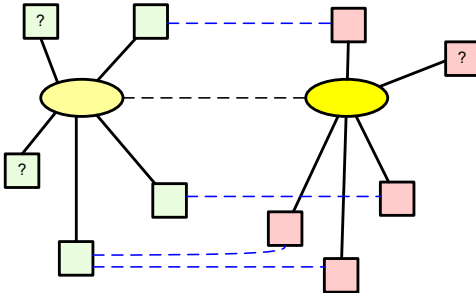
- pmbok:ToolsAndTechniques
- appliesToProcess some pmbok:ManageProjectTeam
- isInputTo some pmbok:ManageStakeholderExpectations
- pmbok:Artifact





Project Management Institute
Poland Chapter

 **Zastosowania**
AGH **Planowane zastosowania 1**


- Budowa odwzorowań pomiędzy klasami PMBOK i Scrum:
 - Rolami
 - Artefaktami
 - Zdarzeniami
 - Procesami
- Wykorzystanie (opracowanie) metod strukturalnego odwzorowania lub metod semantycznych
- Weryfikacja odwzorowań (wyznaczanie odległości pomiędzy klasami)




 Project Management Institute
Poland Chapter

 **Zastosowania**
AGH **Planowane zastosowania 2**

- Skalowanie PMBOK
 - Usuwanie wybranych procesów
 - Usuwanie wybranych artefaktów – uaktualnianie wejść i wyjść
 - Dekompozycja procesów na kilka mniejszych
- Odwzorowanie procesów organizacji w procesy wybranej metodyki
- Definiowanie procesów organizacji na podstawie procesów wybranej metodyki z równoczesnym skalowaniem i dekompozycją
- Integracja metodyki na podstawie integracji modeli ontologicznych

 Project Management Institute
Poland Chapter



Zastosowania

Planowane zastosowania 3 – przykład

Dekompozycja i skalowanie PerformQualityControl

Description: pmbok:PerformQualityControl

- pmbok:Process
- belongsToKnowledgeArea some pmbok:ProjectQualityManagement
- belongsToProcessGroup some pmbok:MonitoringAndControllingProcessGroup
- hasInput some pmbok:ApprovedChangeRequests
- hasInput some pmbok:Deliverables
- hasInput some pmbok:OrganizationalProcessAssets
- hasInput some pmbok:ProjectManagementPlan
- hasInput some pmbok:QualityChecklists
- hasInput some pmbok:QualityMetrics
- hasInput some pmbok:QualityControlMeasurements
- hasInput some pmbok:WorkPerformanceMeasurements
- hasOutput some pmbok:ChangeRequests
- hasOutput some pmbok:OrganizationalProcessAssetsUpdates
- hasOutput some pmbok:ProjectDocumentsUpdates
- hasOutput some pmbok:ProjectManagementPlanUpdates
- hasOutput some pmbok:ProjectQualityManagementUpdates
- hasOutput some pmbok:QualityControlMeasurements
- hasOutput some pmbok:ValidatedChanges
- hasOutput some pmbok:ValidatedDeliverables
- hasToolsAndTechniques some pmbok:ApprovedChangeRequestsReview
- hasToolsAndTechniques some pmbok:CauseAndeffectDiagrams
- hasToolsAndTechniques some pmbok:ControlCharts
- hasToolsAndTechniques some pmbok:Flowcharting
- hasToolsAndTechniques some pmbok:Histogram

hasOutput some pmbok:QualityControlMeasurements
 hasOutput some pmbok:ChangeRequests


hasInput some pmbok:Deliverables
 hasOutput some pmbok:ValidatedDeliverables

hasInput some pmbok:ApprovedChangeRequests
 hasOutput some pmbok:ValidatedChanges

MeasureQuality

ValidateDeliverables

ValidateChanges



Zastosowania

Planowane zastosowania 4 – przykład


Dekompozycja i skalowanie PerformQualityControl

Wydzielono proces **MeasureQuality**

Description: MeasureQuality

Superclasses +

- pmbok:Process
- belongsToKnowledgeArea some pmbok:ProjectQualityManagement
- belongsToProcessGroup some pmbok:MonitoringAndControllingProcessGroup
- hasInput some pmbok:OrganizationalProcessAssets
- hasInput some pmbok:ProjectManagementPlan
- hasInput some pmbok:QualityChecklists
- hasInput some pmbok:QualityMetrics
- hasInput some pmbok:WorkPerformanceMeasurements
- hasOutput some pmbok:OrganizationalProcessAssetsUpdates
- hasOutput some pmbok:ProjectDocumentsUpdates
- hasOutput some pmbok:ProjectManagementPlanUpdates
- hasOutput some pmbok:QualityControlMeasurements
- hasToolsAndTechniques some pmbok:CauseAndeffectDiagrams





Zastosowania Planowane zastosowania 5 – przykład Dekompozycja i skalowanie PerformQualityControl

Wydzielono proces **ValidateDeliverables**

Description: ValidateDeliverables	
Superclasses +	
●	pmbok:Process
●	belongsToKnowledgeArea some pmbok:ProjectQualityManagement
●	belongsToProcessGroup some pmbok:MonitoringAndControllingProcessGroup
●	hasInput some pmbok:Deliverables
●	hasInput some pmbok:ProjectManagementPlan
●	hasInput some pmbok:QualityChecklists
●	hasInput some pmbok:QualityMetrics
●	hasOutput some pmbok:ProjectDocumentsUpdates
●	hasOutput some pmbok:ValidatedDeliverables
●	hasToolsAndTechniques some pmbok:ControlCharts
●	hasToolsAndTechniques some pmbok:Inspection
●	hasToolsAndTechniques some pmbok:RunCharts
●	hasToolsAndTechniques some pmbok:ScatterDiagram
●	hasToolsAndTechniques some pmbok:StatisticalSampling



Zastosowania Planowane zastosowania 6 – przykład Dekompozycja PerformQualityControl

Wydzielono proces **ValidateChanges**

Description: ValidateChanges	
Equivalent classes +	
Superclasses +	
●	pmbok:Process
●	belongsToKnowledgeArea some pmbok:ProjectQualityManagement
●	belongsToProcessGroup some pmbok:MonitoringAndControllingProcessGroup
●	hasInput some pmbok:ApprovedChangeRequests
●	hasInput some pmbok:OrganizationalProcessAssets
●	hasInput some pmbok:ProjectManagementPlan
●	hasInput some pmbok:QualityChecklists
●	hasInput some pmbok:QualityMetrics
●	hasInput some pmbok:WorkPerformanceMeasurements
●	hasOutput some pmbok:OrganizationalProcessAssetsUpdates
●	hasOutput some pmbok:ProjectDocumentsUpdates
●	hasOutput some pmbok:ProjectManagementPlanUpdates
●	hasOutput some pmbok:ValidatedChanges
●	hasToolsAndTechniques some pmbok:ApprovedChangeRequestssReview





Podsumowanie



Podsumowanie 1

- Zbudowano ontologie dwóch metodyki zarządzania projektami PMBOK i Scrum
- Ich podstawowe elementy to:
 - Role
 - Artefakty
 - Zdarzenia
 - Procesy
- Wszystkie elementy są powiązane relacjami pozwalającymi na eksplorację modeli i szybkie wyszukiwanie informacji.
- Planowane są dalsze prace: modelowanie
 - Zmian – w odniesieniu do wymagań, planów, czynników środowiskowych
 - Decyzji – możliwości podejmowania decyzji przez poszczególne role, procesy decyzyjne



Podsumowanie 2

- Zaletą modeli ontologicznych jest:
 - Formalna postać
 - Automatyzacja przetwarzania
 - Zastosowanie maszyn wnioskujących
 - Możliwa jest implementacja oprogramowania do eksploracji modeli i ich modyfikacji
 - Możliwość odwzorowania pojęć wspartych technikami odwzorowania ontologii
 - Możliwość integracji ontologii i przeniesienia jej rezultatów na integrację metodyki zarządzania projektami
 - Na poziomie organizacji (wdrożenie metodyki)
 - Na poziomie ogólnym (proponycja integracji metodyki klasycznych i zwinnych)



Porównanie metodyk zarządzania projektami PMBOK i Scrum przy użyciu modeli ontologicznych – Część 3

- **Dziękuję z uwagą - Pytania?**





Literatura

AGH

1. Marek Jaślan, Branża IT ma się dobrze, www.msipolska.pl/rynek_200702.php4?num=445, 2007
2. Tomasz Sańpruch, www.prnews.pl, Bankier.pl, 2007
3. Tomasz Szetyński, Rynek IT rośnie powoli - firmy, które chcą istnieć na rynku zmieniają swoją strategię działania, www.globaleconomy.pl/content/view/80/17/, 2010
4. The Standish Group, www.standishgroup.com
5. J. Laurenz Eveleens, Chris Verhofer: The rise and Fall of the Chaos Report Figures. IEEE Software, 2010, pp. 30- 36
6. Barry Boehm, Richard Turner, Balancing Agility and Discipline: A Guide for the Perplexed, Pearson Education, 2004, p. 266
7. A Guide to the Project Management Body of Knowledge Fourth Edition (PMBOK® Guide), Approved American National Standard ANSI/PMI 99-001-2008, PMI 2008, s. 460.



Literatura

AGH

8. The state of Agile Development, 3rd annual Survey:2008, Full Data Report Conducted June-July 2008. VersionOne Inc. www.VersionOne.com/acnewsletter, s. 18.
9. Werewka J., Scaling Management of IT Projects Depending on their Size and Complexity, 1st CEE Symposium on Business Informatics, ACS, Vienna 2009, s. 181-190.
10. Mobilizing HP, Project Management as an Executive Priority. Benchmark Implementation: Primavera, Case Study, 2004 Benchmarking Partners, pp. 9
11. Schwaber K., Beedle M: Agile software development with Scrum. Upper Saddle River, N.J., Prentice Hall, 2002
12. Franke, U.; Hook, D.; Konig, J.; Lagerstrom, R.; Narman, P.; Ullberg, J.; Gustafsson, P.; Ekstedt, M.; EAF2- A Framework for Categorizing Enterprise Architecture Frameworks, ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 2009. SNPD '09, pp.327 - 332
13. Gruber T.: A Translation Approach to Portable Ontology Specifications. Academic Press Ltd., 1993
14. Wikipedia: Ontology (information science), [http://en.wikipedia.org/wiki/Ontology_\(information_science\)](http://en.wikipedia.org/wiki/Ontology_(information_science))



Literatura

AGH

15. Rumbaugh J., Jacobson I., Booch G. Unified Modeling Language Reference Manual, The (2nd Edition), Pearson Higher Education, 2004
16. Protégé: <http://protege.stanford.edu/>
17. Jena - A Semantic Web Framework for Java, <http://jena.sourceforge.net/>
18. Euzenat J., Shvaiko P., Ontology matching, Springer Verlag 2007
19. Pinto, H. S., Gómez-Pérez, A., Martis, J. P. Some issues on ontology integration. Proceeding of the IJCAI-99 workshop on Ontologies and Problem-Solving, Stockholm, Sweden, 1999, s. 7.1-7.12.
20. Sowa, J. F. Principles of ontology, onto-std mailing list, Wed, 3 Dec 1997, <http://www-ksl.stanford.edu/onto-std/maillarchive/0136.html>.
21. Kaneiwa K., Iwazume M., Fukuda K.: An Upper Ontology for Event Classifications and Relations. Australian Conference on Artificial Intelligence 2007: 394-403



Project Management Institute
Poland Chapter



Literatura

AGH

22. Allen J. F., Ferguson G.: Actions and events in interval temporal logic. Technical Report TR521, Computer Science Department, University of Rochester, Rochester, New York 14627, July 1994.
23. Cohn M.: Agile Estimating and Planning, Prentice Hall/PTR, 2006, s.330
24. Agile Development Poster, <http://pm.versionone.com/AgilePoster.html>
25. Ivan Terziev, Atanas Kiryakov, Dimitar Mano, Base upper-level ontology (BULO) Guidance, EU-IST Project IST-2003-506826 SEKT
26. Daniel L. Moody, Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions, Data & Knowledge Engineering 55 (2005) 243-276
27. Guarino, Nicola, Chris Welty, Identity, Unity, and Individuation: Towards a Formal Toolkit for Ontological Analysis, W. Horn, ed., Proceedings of ECAI-2000: The European Conference on Artificial Intelligence. Amsterdam:IOS Press. Pp. 219-223. August, 2000.



Project Management Institute
Poland Chapter