

Projekt Interdyscyplinarny

Organizacja pracy

dr inż. Marcin Luckner
mluckner@mini.pw.edu.pl

Wersja 1
3 listopada 2020

Zadania

- Zadania powinny być możliwe do wykonania.
 - nieczytelne, niezrozumiałe zadania.
- Nie powinny zniechęcać do ich realizacji.
 - długotrwałe zadania.
- Powinny być weryfikowalne pod kątem wykonania.
 - zadania bez rezultatu.

Formułowanie zadań

- Zadania powinny być formułowane pełnymi zdaniami.
 - Zwiększamy prawdopodobieństwo, że zrozumiemy o co chodziło w zadaniu.
 - Formułowanie zdania pozwala na konkretyzację i doprecyzowanie zadania.
 - Dzięki pełnej wypowiedzi plan działań złożony z zadań jest dopasowany do konkretnego projektu.

Plan zadań

Plan A	Plan B
<ul style="list-style-type: none">• Napisz UI kalkulatora• Zaimplementuj działania w kalkulatorze• Wykonaj testy deweloperskie kalkulatora• Przekaż kalkulator do przetestowania funkcjonalnego• Przygotuj paczkę dystrybucyjną kalkulatora	<ul style="list-style-type: none">• Zaimplementuj klawiaturę numeryczną kalkulatora• Zaimplementuj wyświetlacz oraz przycisk zerowania w kalkulatorze• Zaimplementuj działania arytmetyczne na kalkulatorze: dodawanie, odejmowanie, mnożenie, dzielenie (dzielenie z dokładnością do 3. miejsca po przecinku)• Sprawdź, czy działania arytmetyczne wykonują się poprawnie; zweryfikuj przypadki brzegowe: działania na liczbach przeciwnych, działania z zerem• Przekaż kalkulator do przetestowania funkcjonalnego• Przygotuj paczkę dystrybucyjną kalkulatora

Rysunek 1: Dwa plany zadań [1]

Długość zadań

- Zadania powinny być nie dłuższe niż połowa dnia roboczego (około 4h).
 - Nic nie szkodzi, jeżeli prawdziwy czas ich realizacji okaże się dłuższy.
 - Ograniczenie ma na celu unikanie tworzenia zbyt ogólnych zadań o trudnym do oszacowania czasie.
 - Krótsze zadania tworzą lepszą perspektywę realizacji projektu.
 - Podobnie jak pisanie całymi zdaniami, skracanie zadań prowadzi do konkretyzacji prac.

Rezultat zadania

- Jeżeli zadanie nie ma określonego rezultatu, albo nie jest on jasno określony to nie można go skutecznie i efektywnie wykonać.
- Rezultat zadania powinien być możliwy do zaobserwowania/zmierzenia.
- Zadania bez rezultatu mogą dowolnie się wydłużać, nie osiągając celu
 - sprzątanie pokoju może trwać dowolnie długo,
 - schowanie brudnych rzeczy do pralki jest weryfikowalne pod kątem ukończenia i można oszacować czas jego trwania (5 minut).

Rezultaty zadań

Zadania bez wyraźnego rezultatu	Zadania z określonym rezultatem
<ul style="list-style-type: none">• Przetestuj serwis ECart• Zaimplementuj dodawanie produktów do koszyka• Przygotuj dokumentację architektury do UI³• Zaktualizuj backlog• Przygotuj tablicę dla zespołu	<ul style="list-style-type: none">• Napisz test dla pozytywnego scenariusza w serwisie ECart opisanego w dokumencie X• Zaimplementuj dodawanie produktów do koszyka• Gdy do pustego koszyka dodam produkt o wartości <code>\$cartValue</code>, wtedy widzę, że koszyk ma jeden produkt i wartość koszyka wynosi <code>\$cartValue</code>• Stwórz diagramy kontekstu oraz modelu (tylko nazwy klas i licznosci w relacjach) i wrzuć je do docs• Wpisz aktualny czas do wszystkich podzadań. Jeśli są „DONE⁴”, to zmień status na <i>closed</i>⁵• Wyklej taśmą na ścianie tablicę z kolumnami: <i>backlog</i>, <i>ready</i>, <i>in process</i>, <i>testing</i>, <i>done</i>⁶

Rysunek 2: Zadania z rezultatem i bez [1]

Cechy wykonywalnego zadania

1. Ma jednoznacznie określony rezultat.
2. Jest przygotowane pod konkretny projekt.
3. Ma krótki założony czas realizacji.
4. Jest sformułowane pełnym zdaniem skierowanym do osoby wykonującej.

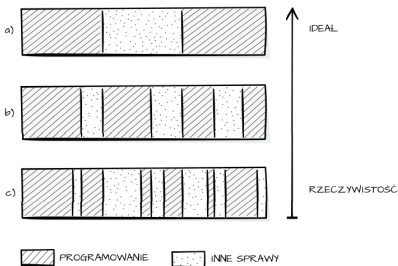
Zadanie w projekcie zespołowym

- Zadanie w projekcie zespołowym ma dodatkowo:
 - Przypisanego jednego wykonawcę.
 - Określony i aktualny status wykonania.
 - Weryfikowalny czas realizacji.
- Zadanie może mieć także przypisane wymagane zasoby, w tym zasoby współdzielone.

Dekompozycja zadania

- Dekompozycja zadania polega na rozłożeniu długiego zadania na pewną liczbę podzadań o mniejszej złożoności.
- Konieczność dekompozycji wynika z współczesnego sposobu pracy, gdy nie można skupić się przez długi czas na wyłącznie jednym problemie.
- Dekompozycja działa także motywująco na poziomie psychologicznym

Dekompozycja, a dzień pracy



Rysunek 3: Dzień pracy [1]

- Praca programisty jest przerywana różnorodnymi innymi zadaniami.
- Długie i złożone zadania wymagają większego nakładu na rozpoczęcie pracy.
- Zanim przejdziemy do realizacji meritum zadania inne zobowiązania wybiją nas z rytmu.
- Małe zadania mają szansę zmieścić się w okienkach na pracę.

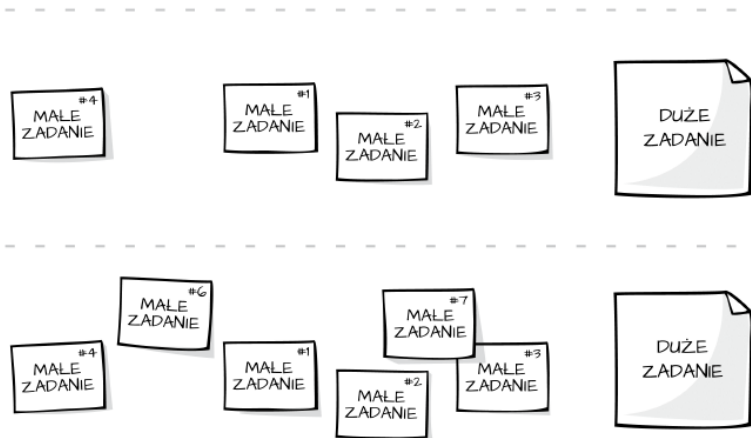
Korzyści z dekompozycji

- Poza dopasowaniem zadań do realiów pracy dekompozycja przynosi szereg korzyści
 - Zmniejsza błąd oszacowania czasu pracy.
 - Ułatwia rozpoczęcie pracy nad zadaniem.
 - Zmniejsza skłonność do odsuwania realizacji zadań.
 - Zwiększa poczucie postępu pracy.
 - Daje możliwość monitorowania postępów.

Mapowanie pracy

- Mapowanie pracy to sposób na zespołowe lub indywidualne mapowanie zadań.
 - Ustalamy duże zadanie, które jest celem do osiągnięcia poprzez małe zadania.
 - Definiujemy małe zadania, które będą składać się na duże zadania.
 - Powinniśmy pracować korzystając z gradulacji czasu trwania zadań.
- Proces mapowania można przeprowadzić korzystając z burzy mózgów.

Proces mapowania pracy



Rysunek 4: Proces mapowania pracy [1]

Pytania

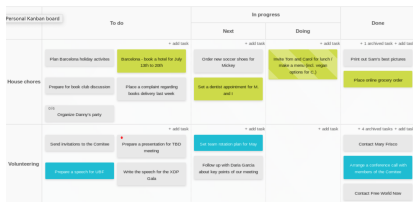
- W procesie mapowania pracy odpowiadamy na szereg pytań:
 - *Co należy zrobić przed małym zadaniem?*
 - *Co należy zrobić po małym zadaniu?*
 - *Czy te wszystkie małe zadania wystarczą, aby wykonać duże zadanie?*
 - *Czy któreś z małych zadań jest nadmiarowe?*
 - *Czy wszyscy rozumieją, co po kolei zamierzamy zrobić, aby wykonać duże zadanie?*

Porządkowanie zadań

- Małe zadania należy umieścić zgodnie z logiką następstw.
- Równoległe zadania umieszczamy jedno nad drugim.
- Gdy duże zadanie nie jest doprecyzowane i trudno jest określić małe zadania należy spisać wątpliwości i przekazać je do odpowiedzialnej osoby.
- Zadania poniżej granulacji łączymy w grupy.
- Zadania powyżej granulacji poddajemy rekurencyjnie mapowaniu pracy.

Kanban

- Kanban jest japońską techniką zarządzania.
- Bazuje na krążeniu kart produktu pomiędzy działami firmy.
- Stosuje się ją przy produkcji oprogramowania do wizualizacji stanu zadań.

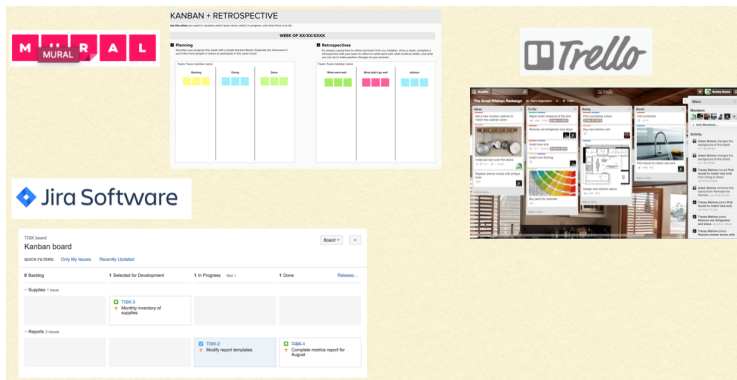


source: <https://kanbantool.com/personal-kanban-board>

Stany

- **To do**
 - Zadania do wykonania w aktualnym cyklu projektu.
- **In progress**
 - Zadania realizowane.
- **On hold**
 - Zadania, które nie mogą być wykonane ze względu na czynniki zewnętrzne.
 - Wymagana interwencja.
- **Review**
 - Zadania zakończone, przekazane do odbioru.
 - Wewnętrzna kontrola jakości (code review).
- **Done**
 - Zadania zakończone i odebrane.

Narzędzia Kanban



Bibliografia I



M. Bartyzel.

Getting Things Programmed.

Helion, 2016.



J. Benson and T. D. Barry.

Personal Kanban: Mapping Work Navigating Life.

CreateSpace Independent Publishing Platform, 2011.