

Ścibór Sobieski

ORGANIZACJA PRACY W PRACOWNICZO-STUDENCKIM ZESPOLE PROJEKTOWYM – STUDIUM PRZYPADKU

1. Wstęp

W lutym 2003 r. na Uniwersytecie Łódzkim podjęto decyzję o organizacji zespołu złożonego z pracowników oraz studentów. Jego zadaniem miało być stworzenie na potrzeby uczelni kilku programów mających docelowo stanowić rdzeń systemu wspomagającego proces zarządzania na UŁ, zwanego dalej Systemem Rektorat.

Już po wstępnej analizie zrozumiano, że docelowy system będzie obszerny i nie da się go zrealizować i wdrożyć w całości, zatem wytypowano programy, które dostarczają najważniejszych informacji dla administracji i w oparciu o nie zaprojektowano jądro przyszłego „globalnego” systemu [Sobieski 2005, s. 273-279]. Jako pierwsze wybrano do stworzenia moduły: kadr, płac oraz rekrutacji [Sobieski 2005, s. 261-271].

Decyzja o podziale całości na moduły miała bardzo duży wpływ na organizację pracy. Podział taki z jednej strony pozwalał na realizację konkretnego modułu w mniejszej grupie – podzespolu, ale z drugiej strony rodził konieczność synchronizacji zadań pomiędzy tymi zespołami.

Tworzenie systemów informatycznych w oparciu o środowisko akademickie wykorzystano również na Uniwersytecie Warszawskim w ramach programu Tempus, przy realizacji Uniwersyteckiego Systemu Obsługi Studiów (USOS) [Mincer-Daszkiewicz 2001, s. 225-236; Mincer-Daszkiewicz 2002, s. 299-314; Sroka 2003, s. 663-674; Mincer-Daszkiewicz 2004].

W niniejszym opracowaniu postaramy się omówić przyjęty sposób organizacji pracy oraz wskazać zaobserwowane jego zalety i wady. Ponieważ organizacja zespołu wiąże się ściśle z przyjętym modelem cyklu życia oprogramowania, więc pokrótce omówimy zastosowany (własny) cykl i odniesiemy się do modeli wzorcowych, znanych z literatury.

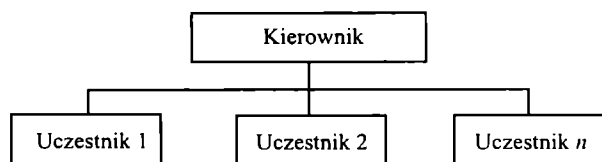
2. Organizacja zespołu projektowego i jej ewolucja

W lutym 2003 r. zespół projektowy składał się z czterech pracowników Wydziału Matematyki oraz jednej osoby – informatyka z Administracji Centralnej. Stopniowo zespół rozrastał się, by pod koniec roku 2005 osiągnąć liczbę 20 osób. W jego skład wchodziły pracownicy naukowo-dydaktyczni oraz studenci. Są to osoby wyłącznie z Wydziału Matematyki UŁ – na tym wydziale od lat prowadzony jest kierunek informatyka, gdzie można znaleźć potencjalnych programistów.

Od samego początku projekt traktowano nie tylko jako „komercyjny”, ale również jako badawczy: komercyjny w tym znaczeniu, że celem nadrzędnym było stworzenie działającego systemu, badawczy, gdyż starano się, by osoby w nim uczestniczące miały możliwość poznania różnych technologii oraz podniosły swoje kwalifikacje zawodowe. Uczestnictwo informatyka, studenta czy pracownika dydaktycznego w tak rozbudowanym projekcie jest niezwykle cenne, gdyż pozwala zweryfikować wiedzę teoretyczną w praktyce. Co więcej, jest to typowa praca zespołowa, której nie można poznać, tworząc programy w jedną czy w dwie osoby.

Należy zaznaczyć, że jednym z bardzo ważnych czynników, które brano pod uwagę przy tworzeniu koncepcji organizacji pracy w zespole, było to, że mamy do czynienia ze specyficzną grupą pracowników. Podczas tworzenia zespołu zakładano od samego początku, że zarówno studenci, jak i pracownicy akademicki będą mogli wykonywać swoje normalne zadania (nauka, prowadzenie zajęć) bez dużych kolizji z pracami projektowymi. To założenie szalenie utrudniło planowanie i układanie harmonogramu, gdyż w pracach brały udział osoby pracujące nieregularnie.

Początkowo jedynie kierownik projektu miał doświadczenie w projektach komercyjnych, dlatego też struktura organizacyjna była płaska, nawet funkcja samego kierownika bardziej sprowadzała się do doradztwa i moderowania, niż stanowiła ścisły nadzór (rys. 1). W początkowym okresie działań istniała możliwość takiej pracy, gdyż grupa liczyła kilka osób i nie było problemu z przepływem informacji i koordynacją prac.

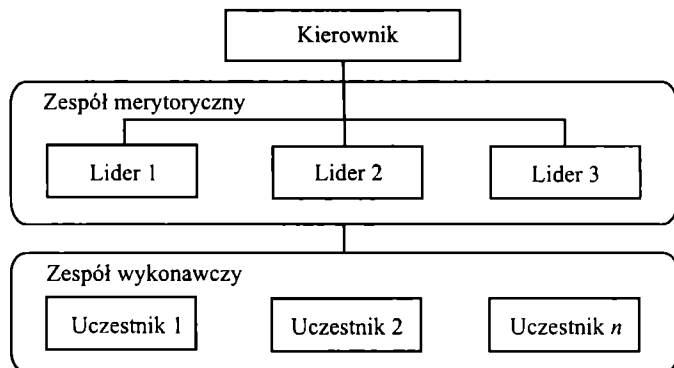


Rys. 1. Początkowa struktura organizacyjna

Źródło: opracowanie własne.

Dość szybko grupa rozrosła się do przeszło dziesięciu osób. Zwiększyło się również doświadczenie uczestników projektu. To pozwoliło na realizację trzech modułów jednocześnie: kadr, płac i rekrutacji. Ostatecznie okazało się, że działania zespołu obejmują prace o bardzo różnorodnej specyfice dla różnych działów admi-

nistracji centralnej. W tej sytuacji stało się jasne, że należy dokonać podziału zespołu pod kątem merytorycznym. W tym celu wyznaczeni zostali liderzy modułów, którzy odpowiadali za ich merytoryczną stronę oraz prowadzili bieżący podział prac i ich rozliczanie (rys. 2).



Rys. 2. Podział organizacyjny z uwzględnieniem nadzoru merytorycznego

Źródło: opracowanie własne.

Powyższy rysunek obrazuje nie tylko samo pojawienie się liderów merytorycznych, ale też pewien specyficzny podział organizacyjny. Omówmy poszczególne grupy odpowiedzialności.

- **Kierownik:** funkcja jednoosobowa, na nim spoczywa odpowiedzialność za wykonanie wszystkich zadań, wywiązywanie się z terminów oraz nieprzekraczanie budżetu.
- **Zespół merytoryczny:** składa się z liderów merytorycznych poszczególnych modułów. Do ich obowiązków należy przede wszystkim nadzór nad prawidłowością realizacji funkcjonalności systemu. Dodatkowo zobowiązani są oni do bieżącego śledzenia postępu prac i przekazywania kierownikowi wszelkich problemów czy potencjalnych źródeł opóźnień.
- **Zespół wykonawczy:** to tzw. zwykli pracownicy, programiści, projektanci, testerzy i inni, których zadaniem jest realizacja zleczanych im zadań.

Tak naprawdę wszyscy łącznie z kierownikiem stanowią zespół wykonawczy i wszyscy mają świadomość, że mogą zostać poproszeni przez inną osobę o pomoc w wykonaniu czynności. Wyjaśnijmy to w kilku punktach.

- Każdy członek zespołu w danej chwili podlega bezpośrednio pod jednego lidera merytorycznego. „W danej chwili” można rozumieć jako „na czas realizacji pewnego większego zadania”.
- Każdy członek zespołu może zostać poproszony przez innego o pomoc w zakresie swoich kompetencji.

- Jeśli może wykonać tę prośbę bez wprowadzenia opóźnień w przydzielonych mu zadaniach, powinien ją wykonać.
- Jeśli następuje kolizja zadań, to lider merytoryczny podejmuje decyzję, czy zadanie dla niego realizowane może zostać opóźnione.
- Jeśli nie może podjąć decyzji samodzielnie lub z drugim liderem merytorycznym, to sprawę rozstrzyga kierownik.

W zespole można i należy wymienić jeszcze jeden podział, nazwijmy go „kompetencyjnym”. Otóż poszczególni członkowie posiadają różnego rodzaju umiejętności i mogą być proszeni o wykonanie prac w ramach tych umiejętności. Wyróżniamy: analityków-projektantów, projektantów-programistów, bazodanowców, testerów, dokumentalistów, przy czym jest to taki podział, że w rzeczywistości wiele osób posiada umiejętności różnego typu, np. bardzo często występuje połączenie projektant-programista, tester-dokumentalista.

Na koniec omówmy przyjęty cykl życia oprogramowania. Zauważmy wpierw, że zespół realizuje kilka modułów jednocześnie, co więcej niektóre moduły mogą być we wcześniejszej fazie tworzenia (analiza, projektowanie), inne zaś – już w fazie utrzymania. Drugim ważnym czynnikiem, mającym wpływ na wybór metody tworzenia systemu, było to, że użytkownik, co prawda wiedział w ogólnym zarysie czego potrzebuje, jednak zgłaszane funkcjonalności nie były wystarczająco precyzyjne. Kolejną rzeczą była świadomość, że tworzone moduły mają być użytkowane przez wiele lat, stąd wiadomo, że wymagania wobec nich (choćby prawne) mogą się zmieniać i należy to uwzględnić.

Ostatecznie postanowiono, że jako metoda tworzenia systemu wykorzystany zostanie zmodyfikowany model przyrostowy [Pressman 2000; Sommerville 1995]. Bezpośrednio z modelu przyrostowego zaczerpnięto cechę tworzenia aplikacji małymi przyrostami w miarę ustalanych z użytkownikiem funkcjonalności. Modyfikacja polega na tym, że stosujemy ten model równolegle do kilku zasadniczych działań jednocześnie. Co więcej, nie ma sztywno określonych zespołów i przypisania osób do modułów (z wyjątkiem liderów merytorycznych).

3. Dyskusja przyjętego rozwiązania

Na pierwszy rzut oka powyższe reguły mogą budzić obawę, że w grupie panuje chaos, gdyż nigdy nie wiadomo, jakie zadanie będzie dziś wykonywała konkretna osoba. Poniżej postaramy się wykazać, że tak się nie stało i prace przebiegają bardzo sprawnie. Wpierw jednak wymienimy zasadnicze czynniki, które miały wpływ na taką, a nie inną decyzję o konstrukcji grupy.

W poprzednim punkcie, przy okazji omawiania przyjętego cyklu życia oprogramowania, wskazano na trzy zasadnicze czynniki, mające wpływ na wybór własnego modelu. Zaznaczmy dodatkowo, że przy wyborze organizacji pracy kierowano się zasadą optymalizacji wykorzystania zasobów ludzkich. Zauważmy, że jeśli przeanalizujemy zarządzanie w zespole tworzącym oprogramowanie według mo-

delu kaskadowego [Pressman 2000; Sommerville 1995], to od razu można wymienić jedną z największych wad tego modelu – to, że zawsze ktoś jest bez pracy. W fazie analizy programista nie ma zajęcia, w fazie programowania rola analityka już się skończyła itd. Stąd też od razu odrzucono próbę realizacji zadań w oparciu o ten model. Podział prac na zespoły jest naturalny w modelu RAD. Jednak i tu nie ma zbyt dużego podobieństwa, gdyż ani projekt nie powstaje w 60-90 dni, ani też nie stosuje się generatorów kodu (aplikacji).

Podkreślmy raz jeszcze, że również w omawianej grupie sztywne przypisanie do konkretnych zespołów zaowocowałoby przestojami osób. Ponieważ pracownicy nie przynależą sztywno do modułów, a każdy wie, jakie umiejętności posiada dana osoba, to może poprosić bądź bezpośrednio ją, bądź jej kierownika o przydzielenie mu tej osoby na czas realizacji pewnego zadania.

Oczywiście powyższe uwagi w dalszym ciągu nie gwarantują porządku w zespole. Obawy mogą budzić dwa ściśle powiązane ze sobą zagadnienia:

- planowanie globalne dla poszczególnych modułów, w tym tworzenie harmonogramu,
- brak kontroli nad tym, co dany członek zespołu robi, oraz to, czy to, co robi, jest niezbędne i ważne.

Aby uniknąć w tym przypadku chaosu, stosuje się hierarchiczny podział kompetencji i przepływu informacji, o czym już pisano w punkcie 2. Omówmy to jednak dokładniej. Po pierwsze, kierownik jest odpowiedzialny za globalne planowanie zadań i przypisanie osób do modułów na czas dłuższy (zwykle tydzień i więcej). Na tej podstawie można stworzyć przynajmniej szkielet harmonogramu. Po drugie, liderzy merytoryczni są odpowiedzialni za śledzenie postępu prac i monitorowanie na bieżąco osób im podległych, przy czym podległość wynika z zasady realizacji zadań, a nie przypisania osoby do zespołu w sensie organizacyjnym. W razie stwierdzenia jakichś opóźnień czy też zapotrzebowania na więcej osób, zgłaszają oni swoje żądania do kierownika. Po trzecie, kierownik wraz z liderami merytorycznymi stanowi komitet sterujący całym zespołem. Co jakiś czas (zwykle co dwa tygodnie) osoby te robią krótkie zebrania i uzgadniają kluczowe plany na najbliższy okres.

4. Zakończenie

Uczciwie należy przyznać się do pewnej porażki organizacyjnej, którą odniósł zespół na samym początku działania. Jak wspomniano w punkcie 2, mówimy o zespole specyficznym, złożonym z osób z uczelni, dlatego też, by osoby te mogły efektywniej pracować i nie musiały przyjeżdżać zbyt często do miejsca pracy, stworzono im możliwość wykonywania swoich obowiązków z domu poprzez Internet [Miodek, Sobieski 2005, s. 75-83]. Jednak takie rozwiązanie okazało się złą decyzją i większość osób dość szybko zrezygnowała z tej formy pracy. Z jednej

strony faktycznie oszczędzano czas na dojazdach i można było pracować w ciszy i spokoju w domu, ale z drugiej strony brak bezpośredniego kontaktu z liderem merytorycznym czy użytkownikiem (praca odbywa się w budynku Rektoratu UŁ) okazał się być na tyle drażniący i utrudniający pracę, że zarzucono ten pomysł.

Podsumowując, uważamy, że wprowadzany sposób organizacji i przepływu informacji w zespole skutkuje dobrymi wynikami i efektywnością. Oczywiście wszystkie działania są silnie wspierane przez narzędzia wspomagające przepływ informacji. Są to programy do śledzenia zmian (błędów, nowych funkcjonalności, modyfikacji istniejącej itp.) oraz program do zarządzania dokumentami, dzięki któremu wszyscy mają dostęp do centralnej bazy notatek ze spotkań, ustaw, rozporządzeń, ustaleń itp. Mimo iż nie udało nam się koncepcja firmy wirtualnej, to wszystkie te programy i procedury przepływu informacji zapewniają poprawne działanie zespołu. Przez dwa lata stosowania omówionej organizacji nie zauważyliśmy powodów do istotnych zmian. Jednocześnie zawsze jeśli zostanie zidentyfikowany problem, konflikt lub wadliwie działająca procedura, natychmiast staramy się wprowadzić jej przemyślaną zmianę. Dzięki temu organizacja wciąż ewoluuje i ulepsza metody postępowania. Najwyraźniej zmiany te widać na przykładzie wdrażanych programów wspomagających zarządzanie informacją.

System Rektorat składa się obecnie z modułów: kadry, płace, rekrutacja, umowy, podatki. Moduł dla kadr został wdrożony we wrześniu 2004 r., płace funkcjonują od stycznia 2005 r., rachunki i podatki – od stycznia 2006 r., rekrutacja w tym roku będzie po raz trzeci prowadzona w sposób elektroniczny. Po takim czasie użytkowania tych modułów można stwierdzić, że udało się stworzyć w warunkach akademickich bardzo sprawnie działający zespół produkcyjny.

Literatura

- Antczak Z., *Organizacja sieciowo-wirtualna a wirtualizacja funkcji personalnej*, Prace Naukowe Akademii Ekonomicznej we Wrocławiu nr 1086, AE, Wrocław 2005.
- Mincer-Daszkiewicz J., *Student Admission System for Warsaw University*, EUNIS 2004.
- Mincer-Daszkiewicz J., *Tworzenie i wdrażanie produkcyjnego oprogramowania w środowisku akademickim*, [w:] *Wybrane problemy inżynierii oprogramowania*, red. J. Nawrocki i B. Walter, Nakom, Poznań 2002.
- Miodek K., Sobieski Ś., *Rozproszona firma w oparciu o Open VPN*, [w:] *Internet 2005*, red. D.J. Bem, A. Kasprzak, M. Szymanowski, t. 1, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2005.
- Pressman R.S., *Software Engineering*, McGraw-Hill 2000.
- Sroka J., *Rozwijanie dużego systemu w warunkach akademickich*, [w:] *Problemy i metody inżynierii oprogramowania*, red. Z. Huzar, Z. Mazur, WNT, Warszawa 2003.
- Sobieski Ś., *Elektroniczna rekrutacja na studia wyższe*, [w:] *Internet 2005*, red. D.J. Bem, A. Kasprzak, M. Szymanowski, t. 1, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2005a.
- Sobieski Ś., *Koncepcja systemu informatycznego wspomagającego pracę państwowej uczelni wyższej oraz omówienie wybranych modułów*, WKiŁ, Warszawa 2005b.
- Sommerville I., *Software Engineering*, International Computer Science Series, Addison-Wesley 1995.

ORGANIZATION OF WORK IN WORKING-STUDENT PROJECT GROUP – CASE STUDY

Summary

The article describes organization of group which created software on requirements of central administrations of University of Lodz. This group consists students and academic employes. This fact, as well as specificity of environment, in which it run, have caused necessity of other look on management of such development group. We try to show in this article advantages of such approaches and we discuss problems that arose during our work.

Dr Ścibór Sobieski jest adiunktem na Wydziale Matematyki UŁ i kierownikiem grupy RPG tworzącej systemy informatyczne na potrzeby administracji centralnej UŁ, e-mail: sci-bor@math.uni.lodz.pl.