

Recenzja rozprawy doktorskiej
mgra inż. Wiesława Kopcia
SPIRAL: A Method of Software Development
with Older Adults' Participation

1. Informacje ogólne

Przedstawiona do recenzji rozprawa składa się z czterech rozdziałów. Pierwsze dwa mają charakter wprowadzający (łącznie 25 stron), rozdz. 3. stanowi zasadniczą część pracy i zawiera 3 artykuły, których pierwszym współautorem jest Autor rozprawy (łącznie 56 stron), a ostatni rozdział jest 2-stronicowym CV Autora. Informacja o wkładzie autora w powstanie prac przedstawionych w rozdz. 3. znajduje się w rozdz. 2.

2. Cel i zakres rozprawy

Głównym celem rozprawy – zgodnie ze sformułowaniem przedstawionym przez Autora – było stworzenie **szybkiej i kompleksowej** metody rozwoju oprogramowania ukierunkowanej na bezpośredni udział osób starszych w tym procesie¹. Zgodnie z intencją Autora, metoda miała być wykorzystywana przez firmy typu **start-up**, które mają **ograniczone zasoby** i możliwości².

Tak sformułowane zadanie badawcze jest bardzo ambitne (z akcentem na „bardzo”) i rodzi pewien problem. Chodzi o to, że prawidłowa realizacja tego zadania wymagałaby nie tylko zaproponowania odpowiedniej metody, ale także wykazania, że proponowana metoda ma pożądane cechy. W pracy jest propozycja metody – jest to metoda SPIRAL (*Support for Participant Involvement in Rapid and Agile software development Labs*) opisana w rozdz. 3.1 – ale nie ma przekonujących argumentów przemawiających za tym, że metoda SPIRAL ma postulowane cechy, tzn.:

- **Jest szybka.** Niestety, w rozprawie nie ma definicji określającej jasno, co Autor rozumie przez szybką metodę rozwoju oprogramowania. Na pierwszy rzut oka podanie takiej definicji może wydawać się trudne, ale jest kilka możliwych podejść. Na przykład można by wykorzystać podejście porównawcze i przyjąć którąś ze znanych metod rozwoju oprogramowania, np. *eXtreme Programming* (w skrócie XP), jako punkt odniesienia (osoba starsza pełniłaby rolę „on-site customer”). Przy tego typu definicji szybkiej metody rozwoju oprogramowania należałoby pokazać, że zespoły o zbliżonym potencjale (tak samo liczne i o podobnych umiejętnościach) działając zgodnie z metodą SPIRAL rozwiążą problem szybciej niż przy

¹ „The key research challenge is the creation of a fast and comprehensive method of software development with direct participation of older adults.” (rozdz. 1.1., str. 12 w drukowanej wersji pracy)

² „The proposed method should be able to be used by emerging technological startups with limited resources and capabilities.” (rozdz. 1.1, str. 13)

wykorzystaniu wspomnianej metody XP. Wiem, że to byłoby pracochłonne – dlatego też napisałem, że zadanie jest bardzo ambitne.

- **Jest kompleksowa.** Tu przydałaby się definicja kompleksowości metody rozwoju oprogramowania ukierunkowanej na udział osób starszych. Pierwsze skojarzenie, jakie przychodzi mi do głowy, to kompleksowość rozumiana jako udział osób starszych we **wszystkich** fazach rozwoju oprogramowania. Ale jak we *wszystkich*, to także na etapie opracowania architektury, kodowania i wdrożenia. Obawiam się, że ten domysł nie jest trafny, bowiem materiał przedstawiony w rozprawie ogranicza się do identyfikacji problemu i współprojektowania prototypu (ang. *co-design*).
- **Wymaga „ograniczonych zasobów”.** To sformułowanie jest nieprecyzyjne i może prowadzić do nieporozumień. Można się domyślać, że Autorowi chodziło o zasoby, które w przypadku metody SPIRAL są mniejsze niż tego wymagają metody LivingLab i OASIS. Jeżeli tak, to należałoby oczekiwać, że Autor przedstawi jakieś dane empiryczne, które za tym przemawiają – niestety, w rozprawie tego nie ma. A sprawa, przynajmniej z perspektywy czytelnika, wcale nie jest oczywista. Metoda SPIRAL wymaga opracowania materiałów szkoleniowych, opracowania i przeprowadzenia gry ulicznej lub czegoś podobnego, a także zorganizowania *hackathonu*. Kalkulując potrzebne zasoby należałoby uwzględnić także fakt, że nie każdy *hackathon* zaowocuje pomysłem, który przerodzi się w produkt komercyjny. Powstaje więc pytanie, ilu (średnio lub przynajmniej orientacyjnie) potrzeba młodych programistów i starszych osób przeszkolić i przekonać do udziału w *hackathonie*, by powstał jeden wartościowy produkt. Jest więc (przynajmniej teoretycznie) możliwe, że gdy mowa o potrzebach ludzi starszych, to metoda *LivingLab* okazałaby się całkiem konkurencyjna wobec metody SPIRAL z punktu widzenia potrzebnych zasobów.

Innym zjawiskiem wpływającym na kosztochłonność metody SPIRAL (a pominiętym w rozprawie) jest **retencja uczestników**. Metoda SPIRAL składa się z sekwencji kroków i opiera się na ochotnikach. Powstaje pytanie, jaka część osób rezygnuje z dalszego uczestnictwa po każdym z kroków. Mówimy o osobach starszych, więc powodów do rezygnacji jest wiele, w tym zły nastrój, choroba, kolizja terminów, czy też zwykłe zapomnienie. To zagadnienie wydaje się istotne, gdyż wpływa ono na liczbę osób, jakie należy „zrekrutować” do pierwszego kroku metody SPIRAL, a tym samym wpływa na zasoby, którymi musi dysponować *start-up*, aby efektywnie zastosować metodę SPIRAL.

- **Ma się nadawać dla firmy typu *technological start-up*.** Najbardziej przekonujące byłyby przykłady firm typu *start-up*, które za pomocą metody SPIRAL zbudowały produkty i te produkty zaistniały na rynku. Jeśli nie, to może chociaż przykłady firm typu *start-up*, które znalazły środki finansowe i ludzi do zorganizowania i przeprowadzenia gry ulicznej lub *hackathonu* dla osób starszych (uczelnia typu PJATK ma zdecydowanie większe możliwości, w tym również autorytet, niż *start-up*, więc zorganizowanie tych wydarzeń przez PJATK nie jest argumentem o szczególnej sile przekonywania). Jeszcze łatwiejszym do uzyskania argumentem (ale mniej przekonującym) byłaby lista firm typu *start-up*, które wyraziły zainteresowanie zastosowaniem u siebie metody SPIRAL. Niestety, żadnego z tych argumentów w rozprawie nie znalazłem.

Mimo przedstawionych zastrzeżeń co do tak sformułowanego celu pracy uważam, że **recenzowana rozprawa jest wartościowa** (a być może nawet *bardzo* wartościowa). By uniknąć wspomnianych wątpliwości wystarczyłoby problem postawiony w rozdz. 1.1 inaczej sformułować. Moim zdaniem lepiej by było zrezygnować z deklarowania szybkości i kompleksowości metody SPIRAL, nie sugerować niewielkich kosztów wytwarzania oprogramowania i nie wspominać o firmach typu *start-up*. Wydaje mi się, że najważniejszą zaletą metody SPIRAL jest dążenie do **upodmiotowienia osób starszych**

w coraz bardziej zcyfryzowanym świecie, w tym danie szansy osobom starszym współuczestniczenia w projektowaniu oprogramowania, czyli pozbycie się poczucia wykluczenia. W tym kontekście metoda SPIRAL bardzo by się nadawała do prowadzenia zajęć w tzw. uniwersytetach III wieku. Tam nie chodzi o biznes, nie musi być szybko, produkty nie muszą się sprzedawać i nie trzeba też sprawdzać, na ile są one unikatowe, czy naruszają czyjeś patenty itp. **Metoda SPIRAL jest bardzo zaawansowaną realizacją podejścia *learning-by-doing*** i z tego punktu widzenia **bardzo mi się podoba**. Oczywiście, to nie wyklucza jej stosowania w innych sytuacjach (w tym biznesowych), ale zwolniłoby Autora z pracochłonnego poszukiwania argumentów na rzecz deklarowanych przez Niego cech metody SPIRAL, które w nowym kontekście przestają być istotne.

3. Wkład Autora

Za najbardziej wartościowe elementy rozprawy uważam:

- **Opracowanie metody SPIRAL.** Składa się ona z dwóch faz: fazy szkoleniowej i fazy „twórczej”. Każda z faz składa się z dwóch kroków. Pierwszym krokiem w **fazie szkoleniowej** jest klasyczne szkolenie z materiałami szkoleniowymi i zajęciami (*blended learning*), a drugim krokiem jest gra wykorzystująca wcześniej zdobytą wiedzę informatyczną i wiedzę, którą z natury rzeczy mają osoby starsze (znajomość historii, w tym historii miasta). Pomysł oparcia szkolenia na grze informatyczno-historycznej uważam za bardzo dobry. Pierwszym krokiem **fazy twórczej** jest generacja pomysłów na aplikację (Autor rekomenduje formę *hachathonu* – str. 54), a drugim – realizacja (łącznie z testowaniem) działającego prototypu (*live mock-up*). Metoda SPIRAL została przedstawiona w rozdz. 3.1 – jest to artykuł opublikowany w materiałach warsztatu „CHASE 18 - Cooperative and Human Aspects of Software Engineering” zorganizowanego pod egidą ACM/IEEE (8 stron).

Zgodnie z deklaracją zamieszczoną w rozdz. 2., Doktorant jest głównym autorem metody SPIRAL i głównym współautorem artykułu na ten temat.

- **Zaproponowanie i współorganizację gry miejskiej (*location-based game*)** mającej na celu przeszkolenie osób starszych z metod korzystania z urządzeń mobilnych. Metoda jest zorientowana na szkolenie przeprowadzane przez młodych wolontariuszy dobrze obeznanych z urządzeniami mobilnymi. Zgodnie z regułami gry, bezpośrednimi operatorami urządzeń są osoby starsze, a wolontariusze pełnią rolę „doradców technicznych” i w razie problemu podpowiadają, co należy zrobić. Gra oraz poczynione w jej trakcie obserwacje zostały przedstawione w rozdz. 3.2. Jest to artykuł prezentowany na konferencji eHealth 360° i zamieszczony w materiałach konferencyjnych w serii LNICST (*Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*), vol. 181 (8 stron).

Doktorant jest współautorem pomysłu na grę, był głównym współorganizatorem gry ulicznej i współautorem narzędzi potrzebnych do realizacji tej gry a także to głównie on analizował dane wynikające z przeprowadzonej gry i jest głównym współautorem artykułu na ten temat.

- **Przeprowadzenie hackathonu z udziałem osób starszych.** W hackathonie brały udział zespoły mieszane składające się z osób młodych i osób starszych. Zadaniem zespołów biorących udział w hackathonie było zaprojektowanie aplikacji ukierunkowanej na osoby starsze. Jest to bardzo ciekawa metoda upodmiotowienia osób starszych w świecie zdominowanym przez informatykę. W trakcie hackathonu poczyniono szereg obserwacji, które mogą się przydać osobom, które chciałyby podjąć się realizacji tego typu przedsięwzięcia. Doświadczenia wyniesione z hackathonu opisano w rozdz. 3.3. Jest to artykuł opublikowany w prestiżowym czasopiśmie *Empirical Software Engineering* w 2018 roku (35 stron).

Doktorant jest współautorem pomysłu na hackathon, przeprowadził analizę danych, dokonał przeglądu literatury i jest głównym współautorem artykułu na ten temat.

4. Wiedza Kandydata

Przedstawiona do oceny rozprawa pokazuje, że Autor posiada głęboką wiedzę nie tylko z zakresu inżynierii oprogramowania (w szczególności eksperymentalnej inżynierii oprogramowania), ale także z innych obszarów, jak np. tzw. srebrna ekonomia (ang. *silver economy*).

To czego mi ewentualnie zabrakło w rozprawie, to przegląd literatury (najlepiej w formie *Systematic Literature Review*) dot. metod wspólnego z użytkownikami rozwoju oprogramowania (ang. *participatory software design/development*). Doktorant odwołuje się do takich metod jak LivingLab, czy OASIS, wspomina też o Google Venture i przywołuje wiele innych prac, ale pominął metodę JAD, która wykrystalizowała się w firmie IBM w latach 70-tych, metodę NGT (*Nominal Group Technique*), czy też elementy partycypacji użytkowników istniejące np. w metodyce *eXtreme Programming*. Literatura na ten temat wydaje się dość spora i zawiera wiele ciekawych obserwacji. Między innymi wart polecenia wydaje się stary artykuł (sprzed ćwierćwiecza), który w bardzo ciekawy sposób porównuje *Participatory Design* z metodą JAD:

Erran Carmel, Randall D. Whitaker, Joey F. George, *PD and joint application design: A transatlantic comparison*, Communications of the ACM; Jun 1993, pp. 40-48.

Mimo tej uwagi jestem pod bardzo pozytywnym wrażeniem wiedzy Doktoranta, którą zaprezentował w rozdz. 1.2, a także w trzech artykułach, na których opiera się rozprawa.

5. Inne uwagi

Praca jest napisana po angielsku. **Jestem pod wrażeniem staranności językowej.** Ogólnie rzecz biorąc, pracę czytało mi się bardzo przyjemnie. Jest jednak kilka spraw (raczej drobnej wagi), które mogłyby być trochę lepiej przedstawione, by ułatwić czytelnikowi zrozumienie prezentowanych treści.

Pierwsza sprawa dotyczy rys. 4 ze str. 53 (ten sam rysunek jest też na str. 20). Nie rozumiem go. Pomarańczowe strzałki sugerują, że proces rozwoju oprogramowania się „zwija”. Ponadto górna „cięciwa” (ta najkrótsza) sugeruje (jeśli nadal poruszamy się zgodnie z pomarańczowymi strzałkami), że po teście jest generacja idei i koniec. Jest to trochę zaskakujące i przydałoby się wyjaśnienie. Nie wiem też, czy nie lepszy byłby inny typu diagramu. Na przykład na osi X można by położyć kolejne fazy rozwoju oprogramowania (np. wg *Rational Unified Process*), a na osi Y pokazać zaangażowanie użytkownika w tych fazach.

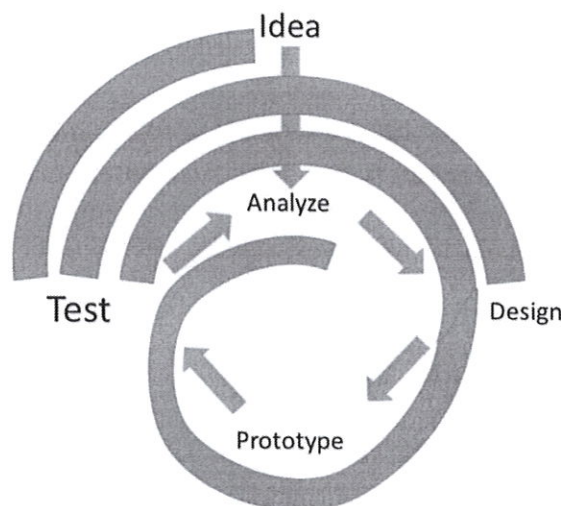


Figure 4: SPIRAL user involvement scheme

Opis metody SPIRAL (na str. 53) wygląda elegancko. Zabrakło mi jednak efektów uczenia się (ang. *learning outcomes*) i związanych z nimi kryteriów wyjścia (ang. *exit criteria*) z poszczególnych kroków – dotyczy to głównie pierwszych dwóch kroków.

Mam nieodparte wrażenie, że w rozprawie zbyt często jest używany termin „*software development*”, gdy tymczasem chodzi o „*software design*”.

Nie rozumiem rys. 2 z rozdz. 3.2 (str. 89 w tym rozdziale). Co oznaczają wartości na osi X i Y (np. wartość „4”)? Autor pisze tylko, że „*The larger values indicate less favorable opinions*”. Nie rozumiem też relacji między rozmiarem kółka a liczbą antonimów. Autor pisze: „*The size of the circles matches the number of antonyms with the corresponding coordinates. For example, the outermost left circle on the left hand side graph corresponds to 3 antonyms for which the median evaluations of the general older other was 2 before the game and 1 after the game*”. Niestety, skrajnie lewe kółko ma rozmiar 1 a nie 3 i zostają w sferze domysłów, jak rozumieć słowo „*matches*”. Nie wiem też, jak odczytać z rys. 2 ile było zmian pozytywnych i negatywnych w odniesieniu do pary *active-passive* albo *pleasant-unpleasant*. Wydaje mi się, że bardziej czytelne byłoby położenie na osi X dwunastu par antonimów i pokazanie na osi Y, ile było dla każdej pary pozytywnych doświadczeń (zmiana z *Negative* na *Positive*) a ile negatywnych (słupek w górę o wysokości 5 mówiłby, że było 5 pozytywnych zmian, a dla tej samej wartości X słupek w dół o wysokości 3 mówiłby, że dla tej samej pary były też 3 zmiany w przeciwnym kierunku).

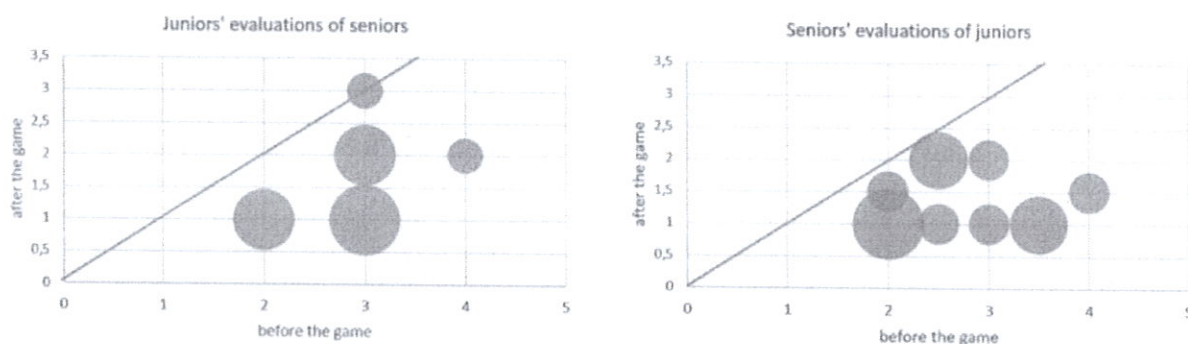


Fig. 2. Median evaluations of general other before and after the game.

W tym samym rozdziale (3.2) na str. 87 Autor pisze: „*Due to limited space we only present the most basic results*”. Szkoda, że Autor nie zamieścił tych danych w rozprawie w formie dodatku.

W rozdz. 3.3 zabrakło mi informacji o informatycznym przygotowaniu osób starszych, w szczególności chciałbym wiedzieć, jak wiele z tych osób uczestniczyło w grze miejskiej opisanej w rozdz. 3.2.

6. Podsumowanie

Uważam, że omawiana rozprawa stanowi oryginalne rozwiązanie problemu naukowego, jakim jest znalezienie metody upodmiotawiającej osoby starsze we współczesnym, mocno scyfrzowanym, świecie poprzez aktywne włączenie tych osób w proces tworzenia oprogramowania. Wstępne rezultaty o charakterze empirycznym są bardzo obiecujące. Zaproponowana metoda SPIRAL wydaje się być szczególnie atrakcyjna w kontekście tzw. uniwersytetów trzeciego wieku. Autor wykazał się dużą wiedzą z zakresu inżynierii oprogramowania, zwłaszcza eksperymentalnej inżynierii oprogramowania. Jestem przekonany, że Doktorant potrafi prowadzić badania naukowe. Tym samym uważam, że spełnione są wszystkie wymagania stawiane rozprawom doktorskim przez ustawę i wnoszę o dopuszczenie Doktoranta do dalszych etapów przewodu doktorskiego.