Badanie wieku i liczby błędów projektów na sourceforge.net

Jacek Pospychała

16 stycznia 2006 roku

1 Wstęp

Badaniu podlega zależność między wiekiem, a całkowitą liczbą błędów projektu (*bugs total*). W dalszej części zostanie zbadany także średni wiek projektów i średnia liczba błędów.
Sourceforge.net istnieje od 1999 roku. Wiek projektu określany jest jako liczba dni od daty jego założenia (*Registered*), do 28 grudnia 2005, czyli daty pobrania próby. Wartości wieku projektu są więc liczbami całkowitymi, w przedziale ok 0 do $(6 * 365) \approx 2000$. W ogólności cecha ta mówi o popularności serwisu sourceforge, a nie wiele o samych projektach. Jej rozkład nie jest znany.
Liczba błędów projektu jest także cechą porządkową, dyskretną. Jednak interpretacja jej rozkładu jest dużo ciekawsza, bo mówi o kondycji projektów. Już pobieżny przegląd zasobów sourceforge wskazuje jednak że projekty najczęściej mają 0 zgłoszonych błędów. Dlatego, by móc lepiej zaobserwować charakter występowania błędów w powiązaniu z wiekiem projektu pod uwagę wzięto także inne cechy, które intuicyjnie mogą mieć znaczenie.
Faza rozwoju aplikacji (*Developement status*) jest bardziej abstrakcyjnym wskaźnikiem wieku projektu. Na sourceforge projekty mogą mieć 7 różnych faz rozwoju. 1-Planowanie, 2-Pre-Alpha, 3-Alpha, 4-Beta, 5-Produkcyjna/Stabilna, 6-Dojrzała, 7-Nieaktywny. Co więcej projekty mogą mieć kilka faz jednocześnie. Jest to więc cecha porządkowa, dyskretna, o nie znanym rozkładzie. Być może jednak wyraźniej odzwierciedli zależność między wiekiem projektu a liczbą błędów.
Aktywność projektu (*Activity percentile*) również potencjalnie może mieć związek z liczbą błędów. Cecha ta służy do tworzenia rankingów projektów na sourceforge i oparta jest na danych statystycznych dotyczących ruchu(liczba pobrań, liczniki odwiedzin), komunikacji (aktywność forum,liczba nowych błędów) i rozwoju projektu(intensywność wydań,cvs commit). Dokładny jej opis jest na stronie `sourceforge.net/docman/display_doc.php?docid=14040&group_id=1`.
Trzeba podkreślić, że cecha ta jest zależna m.in. od błędów. Jednak być może relacja między wiekiem a liczbą błędów projektu występuje tylko wśród projektów o podobnej aktywności. Wartości tej cechy są w procentach. Projekt najbardziej aktywny ma 100%, tysięczny projekt w rankingu ma 99.33%, wartość już poniżej 60% oznacza projekty słabo rozwijane, lub zapomniane, natomiast część projektów nie ma tej oceny (`unrated`). Szersze informacje co do rozkładu nie są znane.
Inną cechą teoretycznie odgrywającą rolę w badanym zagadnieniu była liczba developerów. Jednak pobieżny przegląd błędów w projektach wskazuje, że są one zgłaszane najczęściej przez osoby anonimowe, lub nie związane bezpośrednio z projektem, a nie developerów. Nie byłoby więc realnej korzyści z badania tego parametru, zwłaszcza że przedmiotem badania nie są tylko wieloosobowe projekty. Dlatego ta cecha został pominięta.

2 Próba losowa

2.1 Sposób pobrania próby

W temacie badania zasugerowano pobranie próby conajmniej 50 projektów. Wówczas jednak rozpatrywano jedynie dwie cechy: wiek projektu i liczbę błędów. Wzięcie pod uwagę dodatkowej cechy (aktywność projektu), pod wpływem której część danych być może będzie odrzucana, wymaga aby zwiększyć liczność próby. Brak znajomości rozkładu aktywności projektu nie pozwala oszacować jak duża część próby zostanie odrzucona przez filtrowanie. Najbardziej optymalna byłaby sytuacja, aby po odfiltrowaniu pozostało nadal conajmniej 50 elementów próby. Dlatego podjęto decyzję o pobraniu 100 elementów.
Aby wszystkim projektom zagwarantować jednakowe prawdopodobieństwo dostania się do próby trzeba znać liczność populacji projektów na sourceforge. Niestety dokładna liczba nie jest znana. Sourceforge udostępnia wszystkie projekty w ramach kilku kategorii wyszukiwania K, ze względu na:

- interfejs użytkownika (liczność k_1),

- tłumaczenia (k_2),
- temat projektu (k_3),
- język programowania (k_4),
- system operacyjny (k_5),
- licencję (k_6),
- grupę docelową (k_7),
- faza rozwoju (k_8),
- środowisko bazodanowe (k_9).

Każda kategoria k_i dzieli się na podkategorie k_{ij}, a one na podrzędne kategorie. Kategorie są rozłączne, tzn. nie mają wspólnych podkategorii. Jeden projekt może nie należeć do żadnej kategorii, oraz może należeć do wielu kategorii, lub wielu podkategorii w ramach jednej kategorii (jest pełna dowolność). Np. może działać na kilku systemach operacyjnych, mieć kilka grup docelowych, lub nie mieć ich w ogóle. Liczności kategorii są bardzo zróżnicowane, co ilustruje tabela 1.

Kategoria	Liczność
interfejs użytkownika	70581
tłumaczenia	77585
temat projektu	207352
język programowania	95979
system operacyjny	145864
licencja	75544
grupa docelowa	119737
faza rozwoju	81096
środowisko bazodanowe	8886

Tabela 1: Liczności głównych kategorii

Ponieważ liczności są bardzo zróżnicowane, najlepszym przybliżeniem rozmiaru całej populacji będzie liczność tej kategorii, która ma najwięcej projektów oraz jej podkategorie zawierają najmniej wspólnych elementów (powtarzających się). Na podstawie subiektywnej oceny, do dokładniejszego zbadania pozostawiono kategorie dzielące ze względu na: *system operacyjny* (k_5), *grupę docelową* (k_7), *język programowania* (k_4) i *azę rozwoju* (k_8). Spośród tych kategorii zostanie wybrana najbardziej reprezentatywna (największa) i dająca wszystkim projektom najbardziej zbliżone prawdopodobieństwo dostania się do próby testowej. Ciekawą kategorią jest także *temat projektu*. Jednak trzeba pamiętać że kategorie służą głównie do wyszukiwania i administratorom zależy na wstawianiu swych projektów do największej liczby odpowiadających tematycznie grup. Stąd ta kategoria jest prawdopodobnie mocno nie miarodajna.
Dla każdej z wybranych kategorii pobrano próby po 30 elementów i wyznaczono procent kp_i tych projektów, które występują dokładnie raz w ramach badanej kategorii. Tworząc próby losowano elementy z całej kategorii, jednak nie ma jednakowego prawdopodobieństwa dostania się do próby, ze względu na różną ilość powtórzeń projektów w grupie. Tak więc nie można było wykonać żetelnej estymacji liczby niepowtarzających się projektów, a wyniki odnoszące się do populacji są jedynie naiwnymi szacunkami autora, który musiał w jakiś sposób dokonać wyboru kategorii. Wynik jest w tabeli 2. Dodatkową informację na temat liczby projektów dostarcza sam Sourceforge.net podając liczbę ponad 100 tysięcy projektów.

Kategoria	Liczność	Pojedyńczych w próbie [%]	Pojedyńczych w popul. [liczba]
język programowania	95979	63.333	60465
system operacyjny	145864	30	43760
grupę docelową	119737	30	35922
faza rozwoju	81096	73.333	56700

Tabela 2: Selektywność projektów w wybranych kategoriach. Liczność jest określona na podstawie danych z serwisu, liczba pojedyńczych w próbie na podstawie prób 30 elementowych, a pojedyńcze w populacji, to wartość przypuszczana przez autora ($poj.w.probie \cdot licznosc$).

Na podstawie zebranych danych, zdecydowano się na kategorię *faza rozwoju*. Kategoria ta zawiera ok 56 tysięcy jednokrotnie występujących projektów, a pozostałe występują prawdopodobnie nie więcej niż dwa razy. Automatycznie odrzucone zostały projekty bez fazy rozwojowej, czyli prawdopodobnie nie rozwijane. Rozwiązanie pokrywa się więc z jednym z poprzednich pomysłów, by wprowadzić filtrowanie martwych projektów (a więc zredukować liczbę projektów, gdzie nie są rejestrowane błędy, czyli ich rozwój stoi pod znakiem zapytania).
Zatem dla 100 projektów zostaną pobrane 4 cechy: wiek, liczba błędów, faza rozwoju i aktywność. Grupa liczy 81096 projektów. Losowe numery projektów zostały dostarczone przez generator liczb losowych z www.random.org, czerpiący entropię z szumów atmosferycznych.
Pobranie projektów dalej odbyło się automatycznie poprzez skrypt. Za każdym razem wchodził on na stronę główną kategorii *faza rozwoju*. Wiedząc że ma pobrać projekt numer k, sprawdzał, czy kategoria ma tyle projektów, jeśli tak, zwracał k-ty projekt. Jeśli nie, iterował po kolejnych l-licznych podkategoriach, zmniejszając k o l tak długo, aż $k < l$. Wówczas pobierał listę projektów kategorii i pobierał ten k-ty. Jeśli podkategoria nie zawierała projektów, a podkategorie, powtarzał cykl. Po zebraniu wszystkich stron głównych projektów, odfiltrowano tylko interesujące nas dane przy pomocy `awk`.

2.2 Koszt pobrania próby

W koszt pobrania wchodzą:

1. Napisanie skryptu przemierzającego sourceforge. Czas ok 1 godziny.
2. Czas pobierania projektów - ok 30sec na projekt (przejście przez stronę kategorii, kilka podkategorii i stronę projektu). Razem ok 50min. Potencjalny koszt może być obniżony poprzez równoległe uruchomienie rozsądnej liczby skryptów, zależnie od szybkości łącza jaką dysponujemy.
3. Dodatkowe czynności związane z pozyskiwaniem numerów losowych i filtrowaniem danych. Ok. 10 minut.

Razem proces zabrał więc 2 godziny. Z tego ok. 1 godzina spędzona w internecie, koszt to ok. 3zł wg. cenników kawiarenek internetowych (najtańszy sposób rozliczenia internetu). Jednak ewentualne dodatkowe próby będą już dużo tańsze, gdyż część kosztów była tylko jednorazowa (koszt skryptu). Poza czasem, koszt innych wykorzystanych zasobów był pomijalnie mały. Druga uwaga dotyczy współczynnika aktywności. Zawiera on bardzo znaczące załamanie między projektami poniżej 70%, a tymi powyżej. Takie zachowanie to wg. mnie specyficzna cecha funkcji oceniającej projekty.
Koszt zwiększenia liczby badanych cech z wieku i liczby błędów, do fazy i aktywności jest pomijalnie mały, bo czas potrzebny na ich pozyskanie (10min) jest tylko niewielką częścią całych kosztów.

3 Dane uzyskane z próby

3.1 Szeregi rozdzielcze

wiek	0-120	120-240	240-360	360-480	480-600	600-720	720-840	840-960
liczność	1	8	10	7	3	6	7	9
wiek	960-1080	1080-1200	1200-1320	1320-1440	1440-1560	1560-1680	1680-1800	1800-1920
liczność	5	6	3	4	3	7	5	4
wiek	1920-2040	2040-2160	> 2160					
liczność	3	7	2					

Tabela 3: Szereg rozdzielczy wieku projektów.

l.błędów	=0	1-5	5-10	10-15	15-20	20-25	25-30	30-35	35-40	40-45	45-50	50-55	55-60
liczność	69	22	2	3	1	0	0	0	0	1	0	0	0
l.błędów	60-65	65-70	70-75	75-80	80-85	85-90	$>$						
liczność	0	0	0	0	0	0	2						

Tabela 4: Szereg rozdzielczy liczby błędów w projektach.

aktywność	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100
liczność	4	5	3	3	8	9	6	22	25	15

Tabela 5: Szereg rozdzielczy aktywności projektów

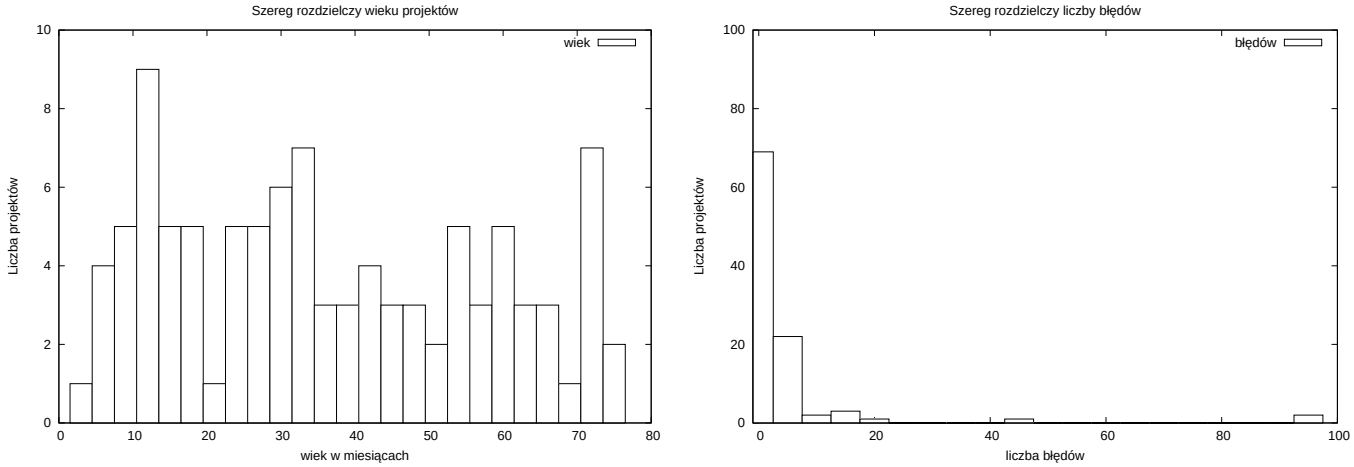
faza proj.	1-Planowanie	2-Pre-Alpha	3-Alpha	4-Beta	5-Produkc.	6-Dojrzała	7-Nieaktywny
liczność	18	19	22	18	18	4	1

Tabela 6: Rozkład fazy projektów

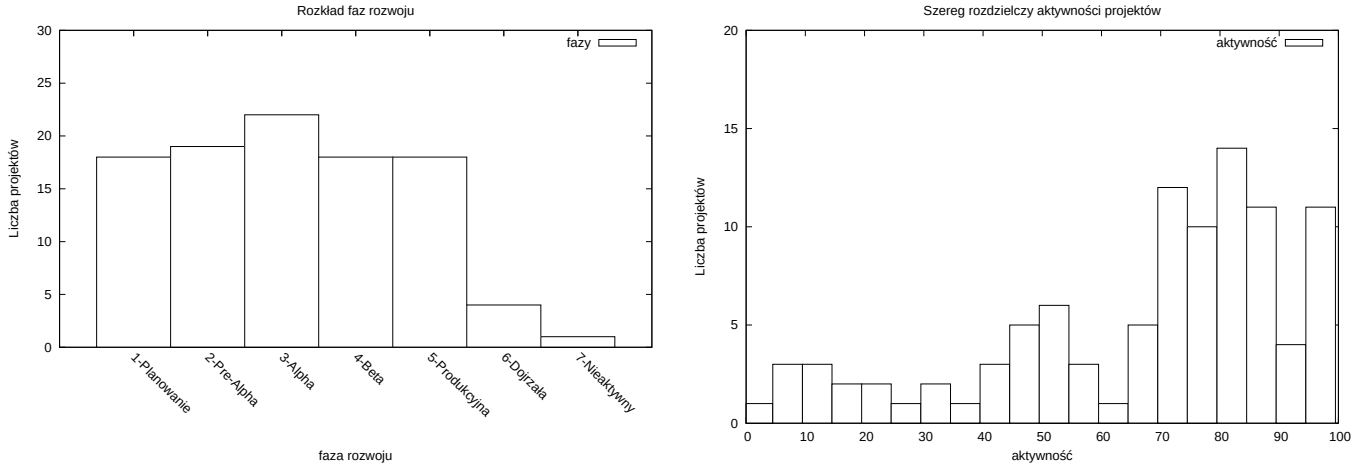
Szerokość klasy w szeregu dotyczącym wieku projektu została dobrana tak by podzielić czas na okrągłe okresy, jednocześnie redukując liczbę klas do ok 20. Najoptymalniejsze okazały się kwartały. W przypadku błędów, pierwszą klasą są projekty bez błędów, stanowiące przeważającą większość. Z drugiej strony, ostatnia klasa na pewno musi być przedziałem otwartym, bo górna granica liczby błędów nie jest znana. Podział szeregu na 20 klas o szerokości 5 został podyktowany tym, by móc wystarczająco dokładnie obserwować najbliższe otoczenie zera, a zarazem ocenić ile projektów "wypada" poza górny przedział (czyli poza 100 błędów) i zalicza się do dużych projektów. Po dwóch cechach, o dużej liczbie klas, tworząc szereg dla aktywności, ograniczono się tylko do 10 przedziałów, gdyż ten procentowy parametr ma dosyć jednostajny charakter, bez gwałtownych skoków. Ostatni parametr, tj. faza projektu ma szczęśliwie tylko 7 różnych wartości, a więc zamiast szeregu przedstawiono pełny rozkład cechy.
Szeregi rozdzielcze badanych parametrów zamieszczono w tabelach 3 - 6. W szeregach pominięto kolumnę częstość występowania, ponieważ dla próby o liczności 100, są to te same liczby co w liczności, z przecinkiem przesuniętym o dwa miejsca w lewo. Na szczególną uwagę zasługuje fakt, że w przypadku błędów, aż 91% ma poniżej 10 zgłoszonych błędów, a 69% projektów mające zero błędów jest nie rozróżnialnych biorąc pod uwagę tylko tą cechę.

3.2 Histogram

Histogramy przedstawiono na rysunkach 1 oraz 2. Co było trudno dostrzegalne w szeregach rozdzielczych, łatwiej dostrzec z histogramów. Nadal nie widoczne są reguły rządzące wiekiem projektów. Liczba błędów koncentruje się wokół zera. Okazuje się że większość projektów na sourceforge tak naprawdę jeszcze nie istnieje, lub nie w pełni gotowa - fazy planowania, pre-alpha i alpha skupiają ok. połowę projektów.



Rysunek 1: Histogramy wieku oraz błędów.



Rysunek 2: Histogramy faz rozwoju oraz aktywności.

3.3 Statystyki opisowe

	aktywność	l.błędów	faza	wiek
średnia	67,31	4,12		1053,41
mediana	74,65	0	3	934
1 kwartyl	52,3	0	2	469
3 kwartyl	85,56	1	4	1588,5
dominanta		0	3	934
wariancja	655,33	386,07	2,35	405111,88
odchylenie std.	25,6	19,65	1,53	636,48

Tabela 7: Statystyki opisowe badanych cech projektów

Nie wyznaczono dominanty aktywności, ponieważ aktywność jest cechą ciągłą. Nie wyznaczono także średniej dla fazy, ponieważ w dyskretnej dziedzinie, w jakiej jest faza trudno interpretować średnią.
Wartość średnia liczby błędów jest pod dużym wpływem wartości skrajnych. Po usunięciu z próby elementu o 169 błędach, średnia liczba błędów wynosiłaby $\overline{X} = 2,41$.

3.4 Liczba błędów a wiek projektu

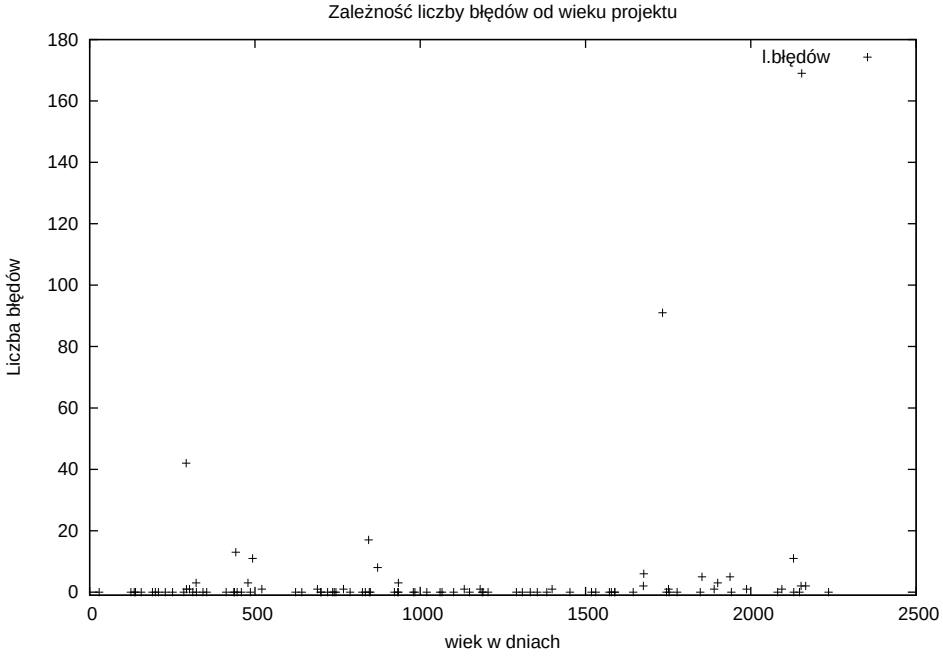
Poglądowe wykresy zależności liczby błędów od wieku projektu liczonego w dniach od założenia, oraz w fazach rozwoju przedstawiają rysunki 3 oraz 4 na stronie 7. Przy posiadanych danych na pierwszy rzut oka widać, że nie ma związku między liczbą błędów a wiekiem projektu ani jego fazą rozwoju. Nie można więc wyznaczyć współczynnika regresji, ani badać korelacji.
Jedyną rzeczą, jaką możnaby w tym przypadku zrobić, to zebrać dostatecznie dużą próbę projektów o nie zerowej liczbie błędów i spróbować określić ich wspólne cechy.
W tym punkcie na chwilę zapominamy o ostatniej posiadanej cesze - aktywności. Nie można mierzyć zależności między aktywnością projektu, a jej wiekiem, bo aktywność jest funkcją m.in. liczby błędów.

3.5 Estymacja średniej liczby błędów i wieku projektów

Do estymacji parametrów: średnia liczba błędów i średni wiek projektu, wykorzystany zostanie estymator właściwości średniej dla dowolnego rozkładu, postaci:

$$\hat{\theta}(X) = \overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{1}$$

Estymator ten jest zgodny i nieobciążony. Przy tym obie cechy spełniają warunki dotyczące skończoności ich wartości średniej w próbie i wariancji w próbie. Wyliczone estymacje wartości średnich badanych cech zawiera tabela 8.



Rysunek 3: Zależność liczby błędów od wieku projektu.

c	$\hat{\theta}_c(X)$
liczba błędów	4,12
wiek projektu	1053,41

Tabela 8: Estymacje wartości średnich liczby błędów i wieku projektu.

4 Podsumowanie

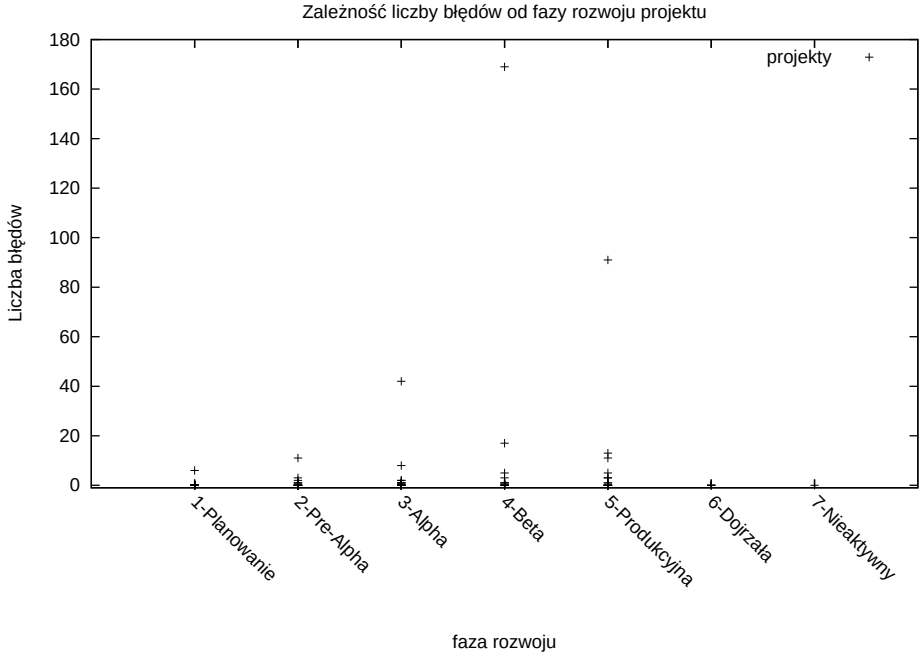
Badanie mające na celu określenie związku między wiekiem projektu, a jego liczbą błędów dało wynik negatywny. To znaczy nie ma związku między wiekiem projektu, a liczbą błędów. Nie ma także związku między fazą rozwoju projektu, a liczbą błędów. Wynik ten odnosi się do całej populacji sourceforge.net. Trudno przypuszczać, żeby oprogramowanie rozwijane na łamach tego serwisu faktycznie było bezbłędne. Bliższe prawdy są natomiast stwierdzenia, że:

1. Rejestrowanie błędów poprzez interfejs sourceforge nie jest popularne. (Na podstawie: brak zgłoszonych błędów w 69% próby, poniżej 5 zgłoszonych błędów w 91% projektów.

2. Nie można jasno, z dużą dokładnością określić liczby projektów na sourceforge. Trudno także powiedzieć ile z nich naprawdę się intensywnie rozwija. Nie mówi o tym liczba zgłoszonych błędów.Cecha aktywność jest nie miarodajna.

3. Wiadomo natomiast że duża część projektów, jest we wczesnej fazie rozwoju. Poza tym, więcej projektów przybyło w minionym roku, niż w poprzednich. Możliwe więc, że serwis sourceforge zdobywa cały czas coraz większą popularność.

Ponadto pobrana cecha aktywność projektów nie została wykorzystana zgodnie z planem. Miała służyć do podziału próby na różne klasy i badania związku wieku i liczby błędów, dla projektów o wysokiej aktywnośći, średniej i niskiej. Ponieważ jednak tylko kilkanaście projektów miało błędy, a na wykresach 3 i 4 nie było widać między nimi żadnych zależności, trudno byłoby dalej filtrować te kilkanaście projektów. Jeden z efektów pracy to pomysł przeprowadzenia dodatkowego badania, które powiedziałoby czym charakteryzują się projekty o dużej liczbie zgłoszonych błędów. Z niniejszych badań wiadomo, że nie podobnym czasem stworzenia, ani etapem rozwoju.



Rysunek 4: Zależność liczby błędów od fazy rozwoju projektu.

5 Dane surowe z próby

Dane z próby. A - aktywność, B - l.błędów, F - faza rozwoju, W - wiek.

A	B	F	W
95,75	91	5	1733
73,33	0	3	2147
76,9	0	4	1383
94,05	0	3	125
93,93	169	4	2154
80,8	0	3	922
85	0	5	251
98,22	0	4	2081
74,65	1	4	1751
67,19	0	3	720
95,84	0	2	135
86,42	0	2	1531
96,54	8	3	871
17,39	0	1	1354
68,42	0	2	984
77,01	3	5	479
96,01	0	5	29
98,96	0	3	229
76,08	0	1	788
51,53	0	6	1453
87,68	0	3	1066
83,47	0	2	623
96,02	17	4	844
92,29	1	4	1181
96,9	42	3	292

A	B	F	W
93,78	11	2	2129
75,41	13	5	442
99,63	11	5	493
80,52	1	3	1133
26,25	0	1	1020
72,49	1	3	689
60,52	0	1	199
83,24	0	2	343
4,67	0	1	1847
54,83	0	1	846
40,07	0	1	486
88,82	0	1	825
9,37	0	2	1644
83,48	0	3	934
54,49	0	1	932
82,1	0	6	1333
81,79	3	5	322
82,2	0	4	446
33,32	0	3	735
19,15	0	3	1291
48,14	0	1	139
74,61	1	3	768
52,85	0	1	1190
79,74	5	4	1852
87,15	0	5	1777

A	B	F	W
76,82	1	5	1987
6,62	0	4	1753
86,55	0	3	2130
10,82	0	5	1589
49,05	2	3	2152
55,65	0	2	745
22,84	0	1	1149
8,5	6	1	1676
11,26	0	4	1578
46,69	0	4	208
44,49	1	3	293
70,57	0	6	699
85,1	0	1	1572
48,21	0	2	156
88,45	1	5	302
72,41	1	2	1889
51,82	1	2	1399
71,78	0	4	1588
24,06	0	1	1101
19,15	0	3	1291
49,57	5	5	1937
35,96	0	3	642
99,28	0	5	190
71,31	0	5	436
58,26	0	2	413

A	B	F	W
67,83	0	5	354
65,98	0	2	285
73,9	1	4	2094
79,65	0	4	979
77,62	2	2	2166
70,19	0	3	740
79,96	0	3	2235
83,54	0	1	834
81,37	0	4	1745
33,98	0	1	700
12,79	0	2	1518
81,04	0	3	1187
82,09	3	4	934
59,22	0	5	312
68,9	0	7	459
72,23	0	4	438
88,53	0	5	849
71,49	0	2	1941
88,69	0	5	1060
98,41	1	2	521
89,71	2	3	1675
86,02	3	2	1900
79,47	0	6	1309
52,78	0	1	1205
43,69	0	2	323