

# Dynamic Process of Internet Companies: An Abstract Model

Leila Kaghazian

Department of Computer Science, University of Southern California

[kaghazia@usc.edu](mailto:kaghazia@usc.edu)

## 1 Introduction

While every 100 days the number of clicks of Internet gets double, the total revenue of e-commerce will grow to up to \$400 billion year in 2001. Not so long ago, it seemed incredible that a Web company could be born in a mere two years. But rather than going back to normal, the pace of creation in Silicon Valley now seems to be speeding up even more. This is the reality of current Internet companies. Any Web company that starts out today and takes two years to get up and running is likely to be left in the dust. In first-generation Internet companies, the founder and a few college buddies moved into a garage. In second-generation Internet companies; the staff coalesces not from friendships but from respect for mutually complementary skill sets. They skip the garage phase, engage two real-estate brokers and make simultaneous bids on three office spaces, hoping one comes through. Some \$301 billion was generated by the Internet economy in 1998, with an annual growth rate over the past four years of 174 percent. The current generations of Internet companies are who want to reduce the schedule of a software development project rather than reducing its cost.

Financing, prototyping, teamwork, marketing, research and development, and the whole process of current Internet companies have evolved in a new model. One of the great aspects of current shift in technology is the dynamic process of Internet companies. Companies with a great success in business have to develop their product as short as possible and make it bug free as much as possible. While the time is very crucial in the whole process, as a typical Internet business site may get 1 million hits in the first day of announcement the notion of traditional prototype may change to a more recent concept "Pre-view". Moreover the average IPO time of successful Internet business is less than 2 years, which is a new phenomenon in software industry. While second generation companies try to keep the IPO inside the company as much they can, the *first mover advantage* factor won't guarantee a successful business unless the company will be really the first mover. Therefore being first in the market encourages Venture Capitals (VC) to put more and more money and to use other resources to put a site up in the market.

CEO of e-companies, a Santa Monica firm which invests money on Internet business as a company factory believe if the Internet world is a yard we are in the first inch. Being in the first step of the new world of Internet the new era of Internet which started a few years ago and going fast had a major effect in science technology, art and industry. In this paper we

briefly to one aspect of this shift in technology: The Process of new generation of Internet companies.

The main purpose of this paper is to provide an abstract model for dynamic process of Internet companies with emphasis on new features on Internet Rapid Application Development. We explain four major factors in IRAD as Outsourcing, hiring personnel, early error elimination and instant bug fixing.

Even though we do have not yet a huge data collection form Internet companies as their keep their IP even in an abstract level for themselves we see how early error detection, outsourcing helps an Internet project to happen in less than 9 month. We also show how sensitive is the model to the number of key factors as personnel, the level of experience, outsourcing, Integration and the number of users who may hit the site in a month.

## 2 Background

There is no doubt that some new concept has defined by second-generation Internet startups. As the barrier of entry against other competitors could be “First mover advantage” every single mistake in development could cost a chapter 11. More over the current generation of Internet startups make *preview version*, a fully running version of future product which may get even 1 million clicks every day<sup>1</sup>.

The main factors, which made an Internet Company model, a radically different with the other software companies, are:

- Speed; the rate of the change of Internet community is fast and more over the current rate also is increasing faster than ever.
- Time: the time to start a new project, develop software or to design a system has been decreasing quickly and there is no chance to make mistakes or learn from experience.
- Development rate
- Human resource
- The wide range of different customer with different taste
- A most likely unlimited budget to do the project

These factors define a new direction in software industry, which we may define a new plan for process of software development and most important one we need to add, modify or define some constraints and new guide lines for Internet Rapid Application Development (IRAD).

---

<sup>1</sup> [www.epinion.com](http://www.epinion.com) is one of the best example of a preview site.

## 2.1 System description

Our model is based on a typical Internet company. Either the company offers an Internet service (like [www.desktop.com](http://www.desktop.com)) or offers a service through Internet (like [www.pets.com](http://www.pets.com)).

The major function of an Internet company is to deliver a service via the Internet to users. The concept of *prototype* here changes to *preview*. *Preview* is a full feature running site, which may, takes more than one million click in a day so it has to be complete version with minimum possible features.

Figure 1 shows an abstract of our system. At the beginning the task will be specified. In second step the whole system get designed and the coding procedure starts after the design.

The shaded are in the model shows our concern about early error detection. So after each step there is a test with a possible rework rate for each step.

As it shows in Figure 1 part of the code send for outsourcing. The major concern for Internet companies are how to manage the site. Load balancing, site crash, platform compatibility and etc is the most concerned issues which force some Internet companies to use outsourcing. Outsourcing helps the management team to focus on their project rather than be worry about these issues.

The integration module is very crucial to integrate the internal and external code. There is a feedback to other part as well as it always need to rework part of the previous work. Internet development faces two major debugging. The first one is instant bug fixing which basically happen when a simple bug in the site may cause a serious crash and has to be fixed very quickly. The second one is long term debugging which it happens when the number of users increase during a given period and the possibility of a bug increase as it pretty much similar to test a software under a huge number of test run.

Internet sites have unique characteristics in which they make a personal unique site for every user<sup>2</sup>. These features make a site interesting but at the same time it may cause some problem in a long run.

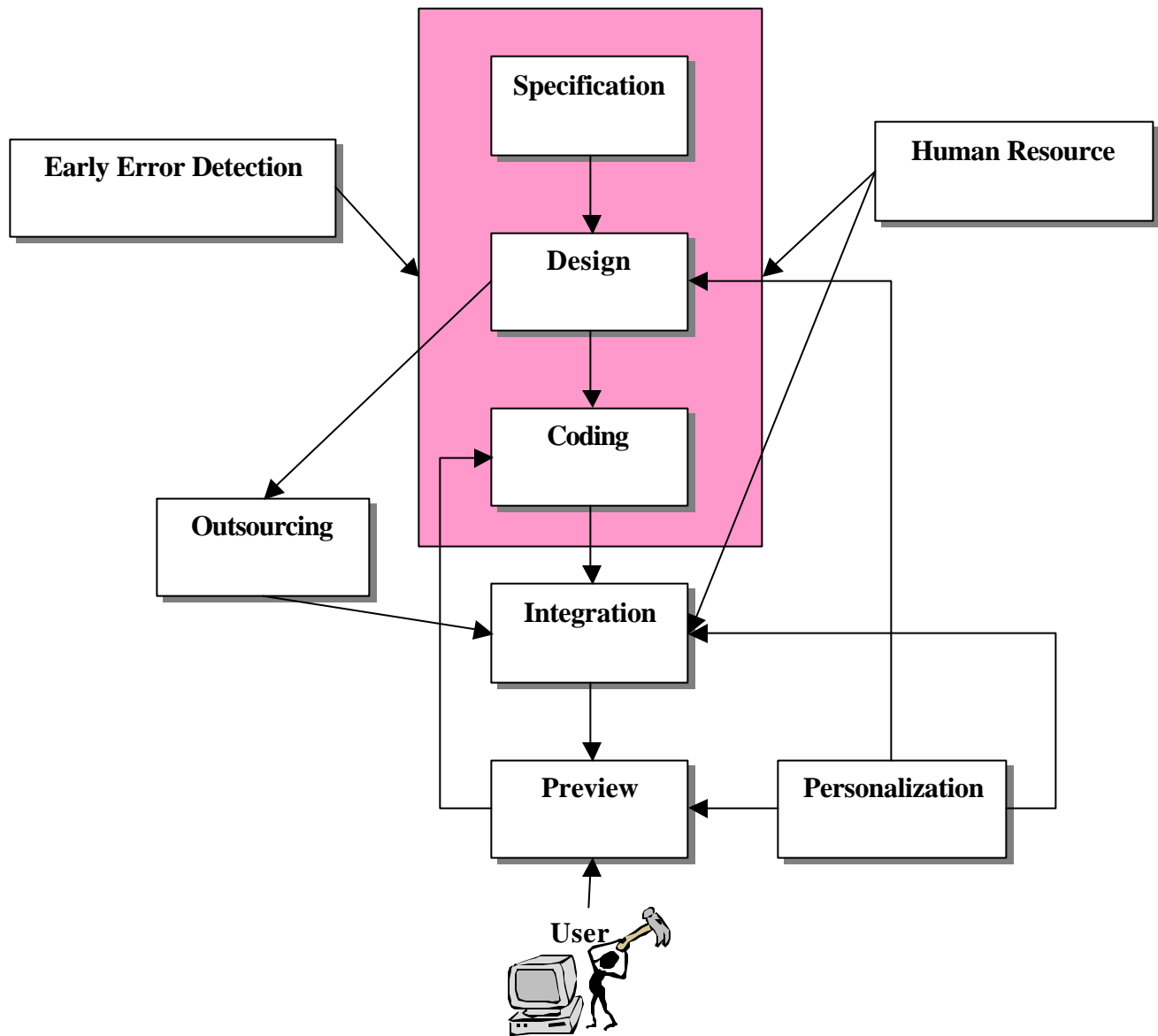
Human resource also is important as it shows in the model this module supposes to take care of staffing strategy in the whole process.

We use of *ithink*<sup>3</sup> software to model for simulation and study our model.

---

<sup>2</sup> For example look at [www.yahoo.com](http://www.yahoo.com) (my yahoo) or [www.cnn.com](http://www.cnn.com) (my cnn)

<sup>3</sup> *ithink* is a registered trademark of high Performance Systems Inc.



**Figure 1**

## 2.2 System reference behavior

Our study shows that several Internet companies has the following facts:

- The time of proof of concept, first round financing and a running site is between 6 to 9 months.
- The second generation use of high experienced engineers from first generation for design test and coding.
- Almost all of the FRAD tree issues are very important in Internet companies.

- Early error detection is very important, and companies do their best to avoid late error detection.
- The size of the development is rather small number around 10 developers.
- Load balancing, site crash and related issues to site maintenance is rather an important issue.

## 2.3 Underlying assumptions

In this model we assume the following assumptions:

- The model is a simple models which only emphasis on part of an Internet Company. It does not cover all aspects of development.
- There are a couple of issues as inspection, concurrency model, group work, resource allocation and management issue etc., which does cover in this model.
- Variables in this model are constant number rather than a complex variable. For example we assume the productivity personnel is a constant rate.
- Our simulation is a closed model. It does not get any input form outside of the model.
- As we used itthink software for our simulation most of the rate are a linear rate unless it specified.
- There is a lot of parameter, which set to nominal values. We found this model from several research notes, papers and from our survey. However these number are open to change to any other given number

## 3 Model development

### 3.1 Modeling process

There is no such a thing as a “rapid-development lifecycle model” because the most effective model depends on the context in which it’s used [1]. So we tried to find the most rapid lifecycle which is compatible with the nature of second generation of Internet companies. After studying most common lifecycles, it seems that IRAD fits in Evolutionary Delivery (ED) lifecycle model plus some modifications. So we used ED+ as a main skeleton of our model.

### 3.2 ED+ Life Cycle

Evolutionary delivery model is a lifecycle model that straddles the ground between evolutionary prototyping and staged delivery. We modify this model to Evolutionary Delivery + (ED+). In ED you develop a version of a product, show it to customer, and refine the product based on customer feedback. The main difference between ED and ED+ is preview vs. prototype and long-term feedback vs. short-term feedback. In IRAD you can not deliver a prototype as you may get 1 million hits in the first day. More over ED+ uses of two different types of feed backs comparing with ED which needs radically different approach to fix.

Figure 2 shows a modified ED which we refer to as ED+.

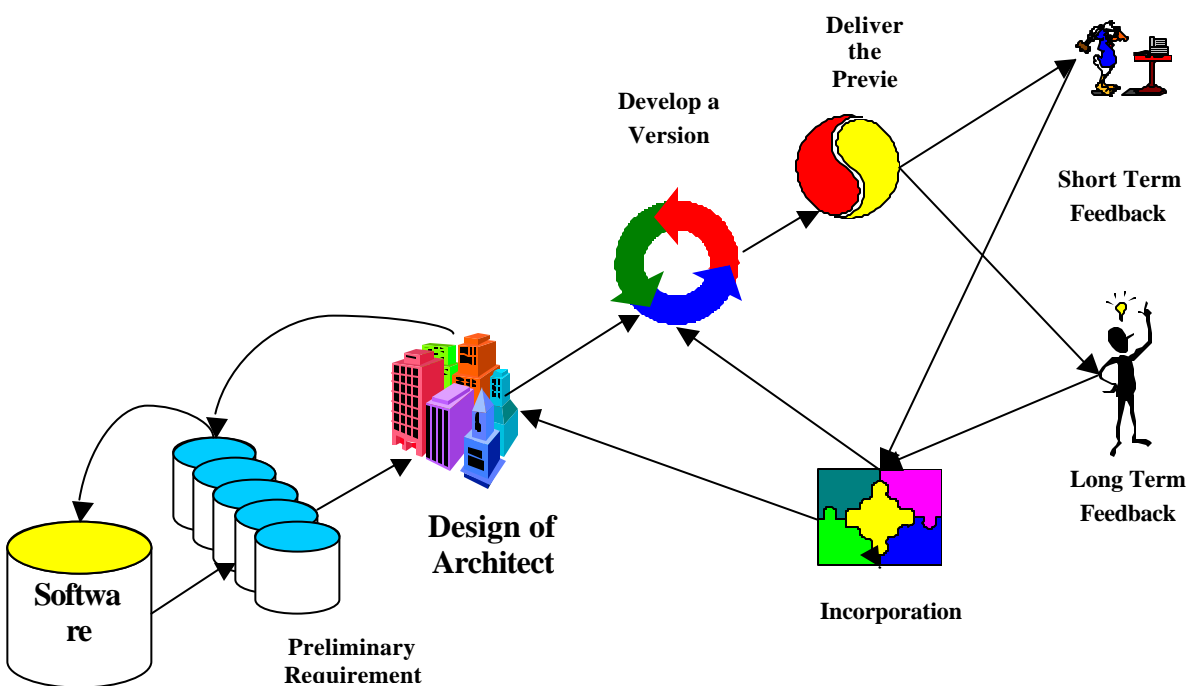


Figure 2

### 3.3 Model evolution: IRAD Features

Barry Boehm in his recent paper[2] “Making RAD Work for Your Project” provides a handful critical factors to make a working RAD. He made an Opportunity Tree for Full - Scale RAD that offers several effective methods for reducing cycle time. We extended the same tree with several new IRAD compatible subdirectories. This new Opportunity tree helped us to improve our model.

Table 1 shows the suggested IRAD enhancemnet to the FRAD Opportunity Tree which are considered as the main features of IRAD.

	<b>FRAD</b>	<b>IRAD Enhancement</b>
<b>Elimination Task</b>	Development process reengineering, Reusing assets, Application generation, Schedule as independent variable	Outsourcing
<b>Reduce time per task</b>	Tools and automation, Work Streamlining (8-20), Increasing parallelism	Use of outsourced application and services as temporarily means to increase parallelism
<b>Reduce risks of single-point failure</b>	Reducing failures, Reducing their effects	Fall back scenarios to cover the lowest level of the service when the higher level failed
<b>Reduce backtracking</b>	Early error elimination, Process anchor points, Improving process maturity, Collaboration technology	Use of the first generation companies to do correct by design
<b>Streamline activity networks</b>	Minimizing task dependencies, Avoiding high fan-in, fan-out, Reducing tasks variance, Removing tasks from critical path	Use of temporarily outsourcing to achieve the above
<b>Increase the effective workweek</b>	24 * 7 development, Nightly builds testing, Weekend warriors	Stock options as powerful incentives

**Table 1: IRAD Enhancement to FRAD**

In addition to above-mentioned enhancements, there are some features, which we consider especially for IRAD. These items has shown in Table 2:

<b>Feature</b>	<b>Description</b>
<b>Preview vs. Prototype</b>	A stable version with only water down basic features Helps to educate the users before the full feature is up
<b>Rapid market adaptability</b>	Offer much room for creativity More room to put bugs as features!
<b>Importance of front end vs. backend</b>	Mistakes and low quality front end and unforgivable Much more tolerance to the backend: Backend can fit better to the classic software development pradigm
<b>Highly heterogeneous environment</b>	All different browser should be covered from day one
<b>Instant update</b>	Classic version update release of the software is obsolete Instant update of all users with no lag Avoid download of much software in the user side Better control over test of new ideas Instant debugging
<b>Personalization</b>	Key to a win-win situation for the vendor and user
<b>Mass customer support</b>	Preview asks for mass customer support from day one

**Table 2: Special IRAD features**

### 3.4 Major IRAD factors

Considering the main IRAD features we select the following factors as the main issues for our model . These factors are as follows;

- Outsourcing
- Hiring personnel
- Early error detection
- Instant debugging

In next 4 subsections we explain each factor in more details.

#### 3.4.1 Outsourcing

A couple of years ago, e-commerce was just knocking at the door of the Information Technology (IT) market. Today, companies have shoved that door open and are walking through to discover new opportunities and new ways of doing business more profitably and more efficiently. If the industry prophets are to be believed, many others will join those companies as customer demand drives e-commerce from an option to a necessity for success. In the new era of IT industry outsourcing might be a big factor for success.

Outsourcing is expected to play a major role in e-commerce. Stan Lepeak, an analyst with Meta Group, predicts that virtually all organizations will be outsourcing some component of e-commerce. Many organizations will outsource the entire operation. What you, as a company, need to decide is what part of your e-commerce operation is suitable for outsourcing and what part, if any, you want to save.

Based on a survey [3], top 10 Reasons Companies Outsource are:

1. Reduce and control operating costs
2. Improve company focus
3. Gain access to world-class capabilities
4. Free internal resources for other purposes
5. Resources are not available internally
6. Accelerate reengineering benefits
7. Function difficult to manage/out of control
8. Make capital funds available
9. Share risks
10. Cash infusion

This report also shows that top 10 Factors in Vendor Selection are:

1. Commitment to quality
2. Price
3. References/reputation
4. Flexible contract terms
5. Scope of resources

6. Additional value-added capability
7. Cultural match
8. Existing relationship
9. Location
10. Other

Furthermore, the report indicated that top 10 Factors for Successful Outsourcing are as follow:

1. Understanding company goals and objectives
2. A strategic vision and plan
3. Selecting the right vendor
4. Ongoing management of the relationships
5. A properly structured contract
6. Open communication with affected individual/groups
7. Senior executive support and involvement
8. Careful attention to personnel issues
9. Near term financial justification
10. Use of outside expertise

Beside the above factors and rules about successful outsourcing, one should consider the software outsourcing costs which means the trade off she accepts with outsourcing:[4]

1. Additional expenses
2. Additional contractual and organizational overhead
3. Bait and switch – vendor substitutes less qualified personnel for those originally specified
4. Loss of intellectual property

Then you face another set of questions. What vendors have skills that meet your needs? Do you want to use one full service vendor or multiple vendors? If you go the multiple vendor routes, do you want to manage the relationships or outsource the management? Consider those questions carefully, because the answers can determine whether or not your e-commerce experience is a success. Thus, even though outsourcing is a huge alternative for rapid development it comes with a relatively high risk.

In our study on dynamic process of Internet Companies we put the outsourcing as one the main module in the whole process of software development. To be more specifics recognized Internet companies use outsourcing to put their software on the net and they focus on their service rather than load balancing, site crash etc (They try to keep their intellectual properties as much as possible inside the company).

In our model we supposed these companies select the right vendor and also provide an acceptable interface to negotiate with the vendor. Based our data, most of the Internet companies outsource about 5% to 10% of their work to accelerate the project.

### 3.4.2 Hiring Personnel

Based on our data, usually the second generation Internet startup companies are consisting of 15 to 18 personnel. About 8 to 10 person of these people are involved with technical issues. To follow the Rapid Application Development, these rules should be considered:

- Going from low spending to medium spending on staffing, training, and work environment produced proportionate gains: additional spending was justified on roughly 1-to-1 payback basis. But when spending on staffing, training, and work environment went from medium to high, productivity skyrocketed, paying back 2 to 1 or 3 to one[1]. So Internet companies should hire top talent personnel as much as possible.
- Hiring job matching staff [4]
- Select people who will complement and harmonized with each other (Team balance)
- Eliminate and replace problem team members as quickly as possible (Misfit elimination)
- Help people to self-actualize rather than forcing them to work
- You must have fewer than 8 to 10 people for a team to jell [5]

### 3.4.3 Early Error Detection

One of the main suggested approaches in IRAD is early error detection. We refer to this as "correct by design". It means in each step like Specification, Design, etc. a major error detection task will fire to make guarantee a fairly bug free outcome. A wrong design or a simple error may cost losing one million customers in the first year of launching the site and even sometimes it makes an impossible comeback for a company. In addition there are several rules about early error detection like

“A design error left undetected until system testing typically takes 10 times as long to fix as it would if it were detected at design time[1]”

or

“Reworking defective requirement, design and code typically consume 40 to 50 percent of total cost of software development[1]”

That made us to emphasis on early error detection in our model.

### 3.4.4 Instant Debugging

The main difference between ED and ED+ is the notion of "instant bug fixing". In a normal software company the time difference between releasing a new version and getting customer feedback could be one week, one-month etc. But in Internet world the

time difference is a "click". Instant bug fixing could save a lot of money and could increase in the level of software stability and acceptability by users.

### 3.4.5 Other rules and factors...

There are some rules and factors, which we paid attention to them in our model. Here are some of those:

- Take advantage of electronic mails, electronic bulletin boards, Intranets and etc. reduces the classic communication overhead.
- Shortcutting 1 day of QA activity early the project is likely to cost you from 3 to 10 days of activity [1].
- Getting a requirement right in the first place typically costs 50 to 200 times less than writing until construction or maintenance to get it right [6]
- Even if you are successful at avoiding requirement gold planning, the average project experiences about 25- percent change in requirements over its lifetime [1].
- About 40 percent of all software errors have been found to be caused by stress; those error could have been avoided by scheduling appropriately and by placing no stress on the developers [6]
- Break the schedule into three, sometimes four milestones, which represented completion or stabilization points for major feature clusters

## 3.5 Data acquisition

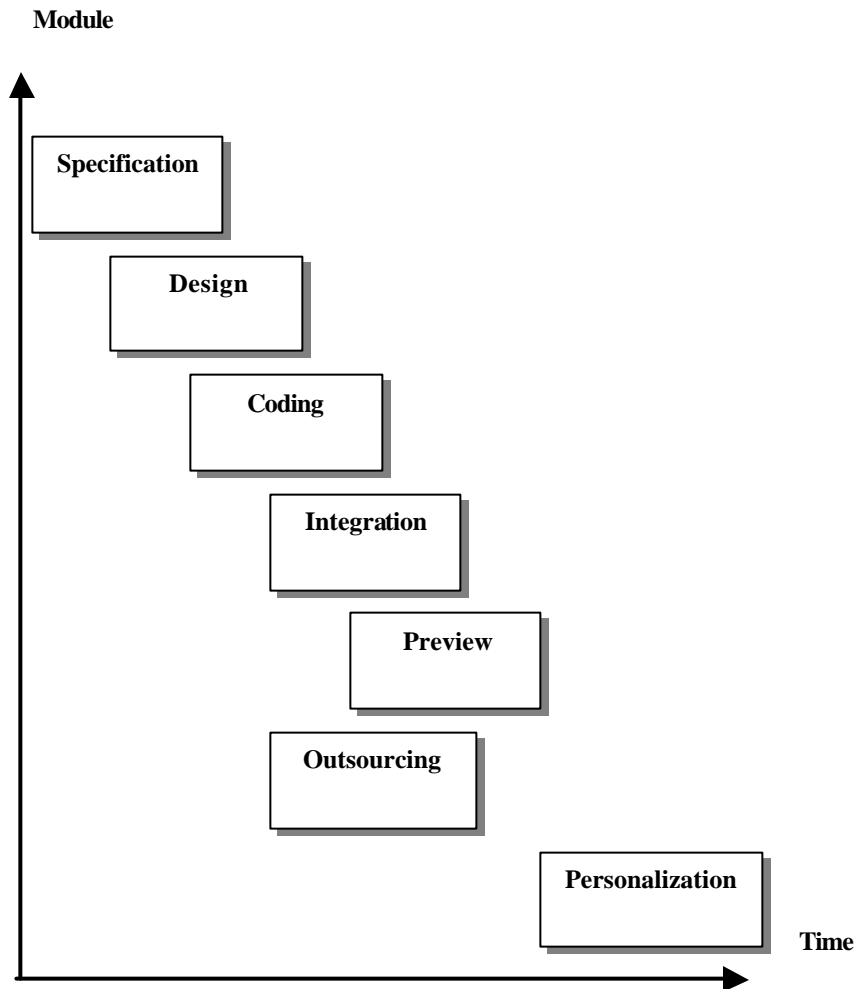
In order to collect data, first we sent an e-mail to most recent Internet companies, which we couldn't get any information on that way. Secondly, we interviewed several per-IPO or on going companies CTOs and VPs. Then we prepared a questionnaire (Appendix A) and send it to 20 different Internet companies CTOs (we've just received two questionnaire back up to now). We also studied several books and papers to get more scientific data about our project.

The problem is that we suffer of lack of real data that is really necessary to evaluate and improve our model. However we could interview a couple of CTO's of first round financing and second round financing Internet companies. These companies are [www.u4cast.com](http://www.u4cast.com), [www.values.com](http://www.values.com), [www.lawyerconsult.com](http://www.lawyerconsult.com) .

The next step is to distribute questionnaire to more companies and to tune the model and assumptions.

### 3.6 Model Timeframe

Figure 3 shows the system time frame . As it shows in this figure the human resource module in Figure 1 try do distribute the load of development over the whole process.



**Figure 3**

## 4 Model Description

In this section we describe the five components of an IRAD model. These components are *Specification and Design*, *Out Sourcing*, *Development*, *Integration and Personalization* and *Human Resource*. In the following sections we describe each part with in more details. We used itthink<sup>4</sup> software as the main tool for simulation, design and analysis of this project. Fig.4 shows model description in an abstract format designed by itthink software.

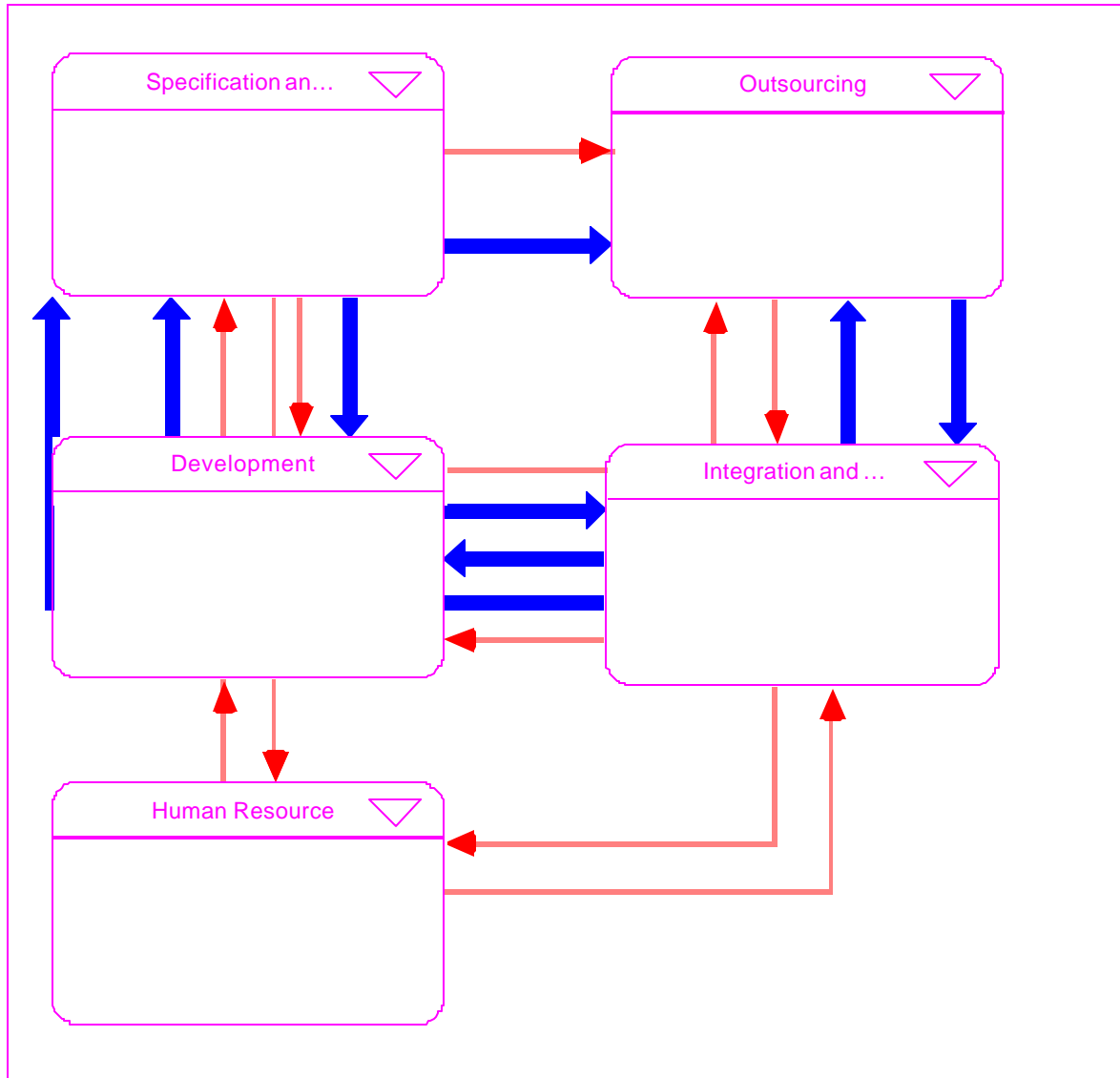


Figure 4

## 4.1 Model Assumptions

There is a couple of general assumption we used in this model . In the following we point out to these assumptions:

- ***Productivity(function point/person-month)***

We assume the nominal specification productivity and design productivity and site design productivity for a person as 2 *function point/person-month* as we consider engineers as senior expert

For coding, Integration, 1.3 as productivity as we may hire not very experienced engineer for coding and integration. .

- ***Test***

We assume the amount of test for each section is 10% of the whole work. This number could be change for sensitivity analysis.

- ***Local Rework Rate***

We assume a local rework error rate for all of the stage. We assume the nominal rate as .01. User can change this number for test

- ***Outsourcing delivery rate***

We assume .8 function point / month for outsourcing delivery rate. For more explanation please look at outsourcing part.

## 4.2 Specification and Design

Fig. 5 shows the specification and Design with emphasis on *early error detection* module in our IRAD model. As we see in Fig.5 the main concern in IRAD is to detect any error in early stages. Even though it seems to be time consuming but the importance

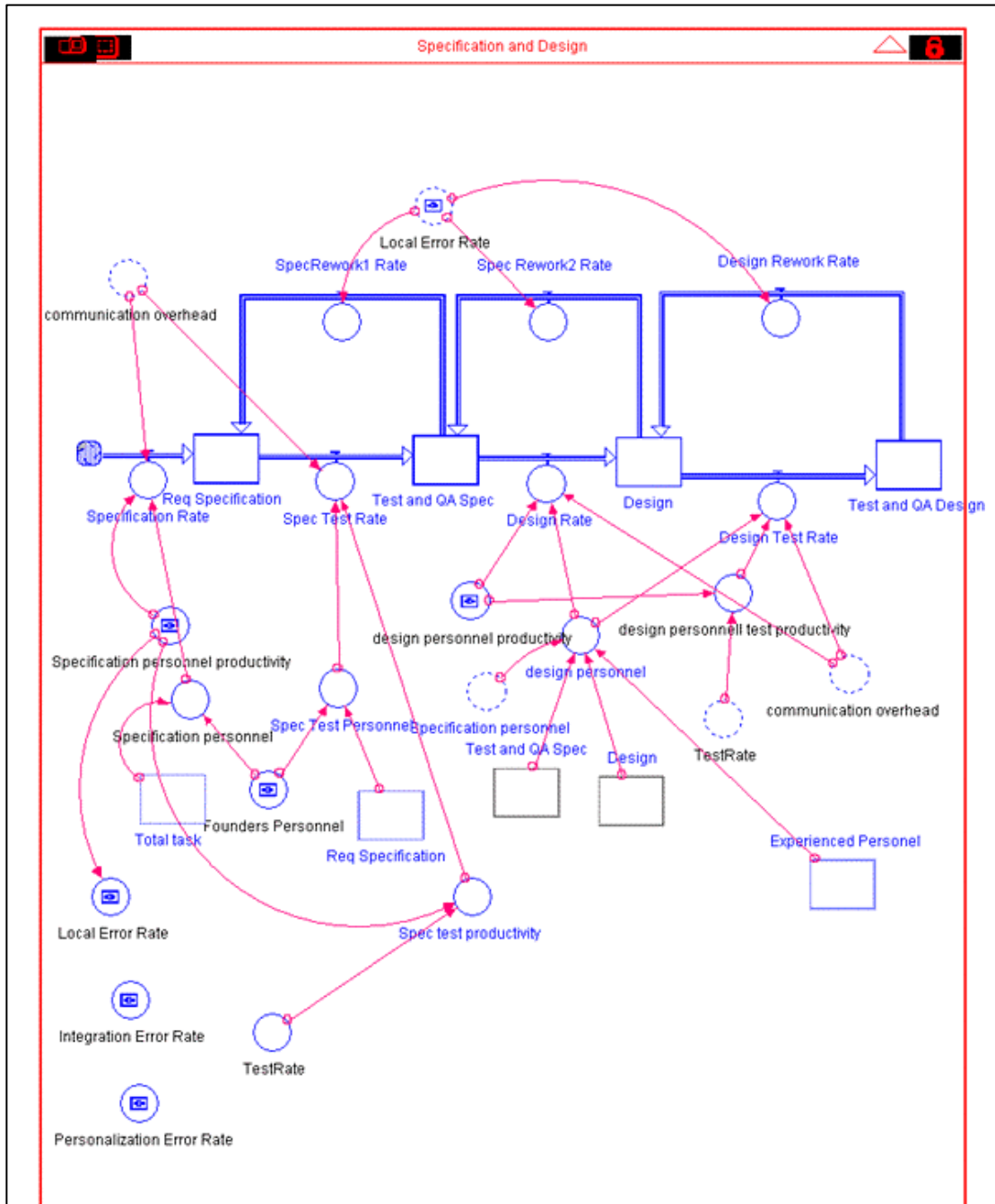


Figure 5

of a bug free preview and the fact that a company has only one chance to do such experience will force an IRAD model to detect any error as soon as possible in the evolution process. In the following we explain the process and each component in more details.

- Total Task: we assume about 14- 16 major function for a Internet company. The assumption is based on our finding form a CORDOMO and interview with a few CEO of software companies.
- Req Specification(function point): It shows the specified task. The rate of specification depends on the specification productivity. For a normal employee we assume a rate of 1. For a founder of a second-generation company with specific background in Internet field we assume a rate about 1.4, and for other employees this rate is between 1.2 and 1.4.
- Test and QA Spec(function point): This receiver indicates the importance of a test after specification. Our assumption is that, founders do the test to make sure there is no bug in specification.
- Design(function point): After test and quality assurance specification the founders and a team of new hires start the design of the software.
- Test and QA Design(function point): Same as previous stage there is a test after the design. E show in the result section that how important is the accuracy of the design and specification.
- Rework Rate: As it shows for each step we consider a rework factor in which part of the work may redo by engineers. We assume a *local rework rate*, for each step. We consider a similar rate for all stages without losing generality, however we can have a different rate for each stage. *Local rework rate* depends on the level of seniority of founders and experienced personnel. The more experienced an engineer the rate of rework decrease. The rate of *local rework rate* could be changed for the sensitivity analysis to this factor.
- Long term feedback: As we explained in ED+ in Figure 2, one of the main difference between ED and ED+ is the long term feedback from Integration and Personalization to Design and Specification. As there is direct flow from Personalization to Integration we just consider a flow from Integration to Design and Specification. We consider this factor independent of engineer's knowledge as the rate of change in Internet world changes, so there is no way to avoid of such rework and feedback.

### 4.3 Out Sourcing

Outsourcing is the second most important feature on IRAD. As we may see in this module outsourcing considers as a major factor of development. While the negotiation

rate is very important and could be the major concern of outsourcing time, the integration process plays an important rule in the whole process of development.

The third main element of IRAD model is instant bug fixing which may get combined with personalization. More people visit the site the probability of getting an error will increase until it gets stable. However stability won't stay long as developers add more features and change configuration every month or every quarters.

Figure 6 shows the outsourcing part of the model. We explain in more detail each component.

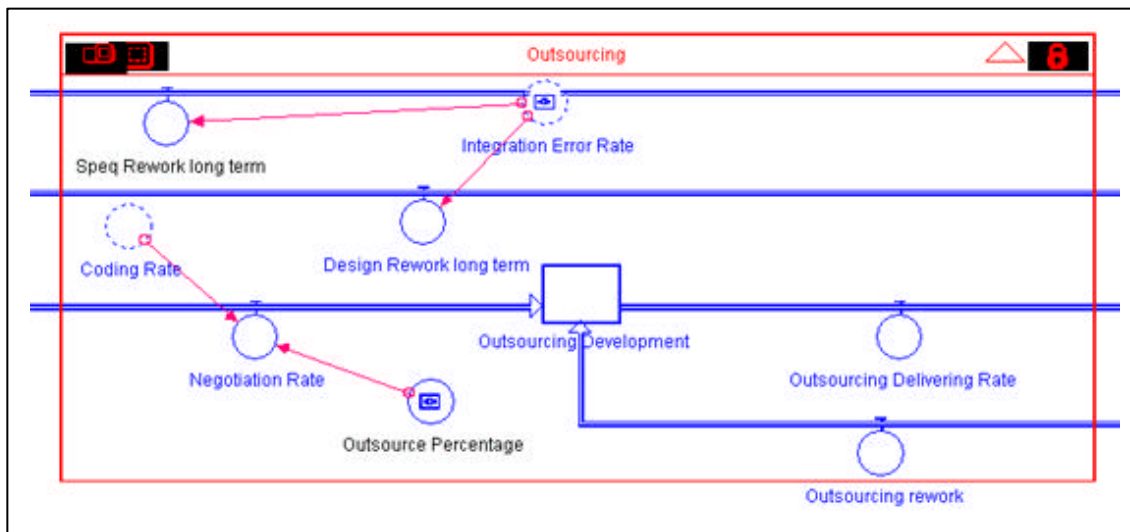


Figure 6

- Negotiation Rate(function point/month): This indicates what percentage of the whole coding goes for outsourcing. If the negotiation drop by any reason basically the whole work will be done in the company.
- Outsourcing Percentage: This is depends on the company management to decide how much is the work may goes for outsourcing. Normally we found between 5% to 10% goes to for outsourcing without losing the IP values. The negotiation rate is the multiplication of Coding rate and Outsourcing percentage to make sure only part of the work goes for outsourcing.
- Outsourcing Development(function point): This shoes the amount of work outsourced by the third party.
- Outsourcing Delivering rate(function point/month): We assume the third party deliver the work with an acceptable rate. There is common belief that outsource vendor rate is basically less and about the half of a developer. But as the number of developer and the available resource in the vendor is enough to deliver we assume just an acceptable rate.

- Outsourcing rework: We assume rate of rework for outsourcing. Generally it depends on the quality of the vendor and we already addressed the major factor to select a vendor for outsourcing to decrease in this rate.

We do not consider any more details about the outsourcing and we leave this detail to the vendor and it is beyond the context of this paper.

#### 4.4 Development and Integration

Development and Integration consist of two major part: coding and Integration. Integration without any doubt is most time consuming part of an Internet rapid development application, especially with outsourcing. We consider the same group works for coding and test. We explain each component in more details:

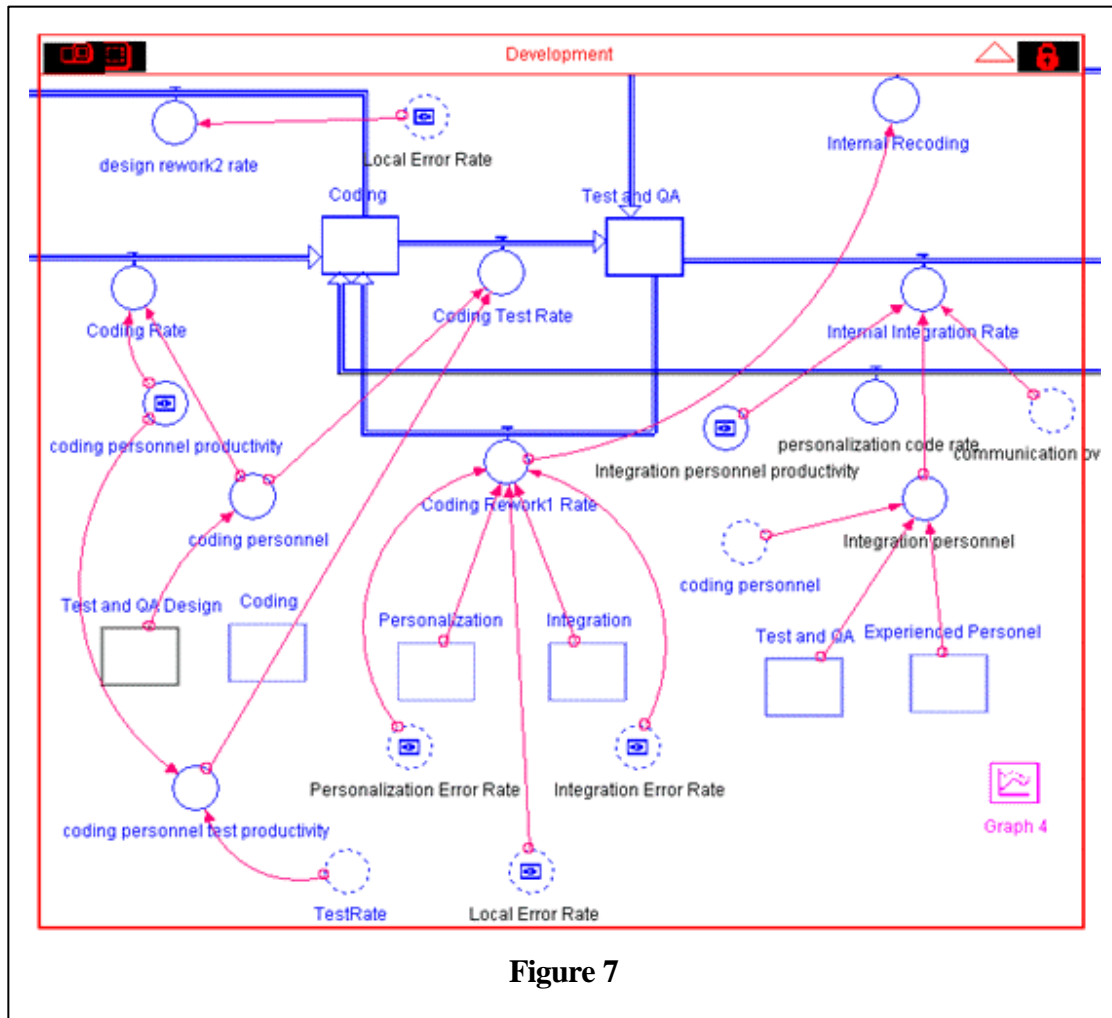


Figure 7

- Coding as it shows in the Fig 7 there is two major feedback to coding. First from test as local rework and second from personalization. The main idea is that as the competition goes high every quarter there are some features, which has to add to the

site. These features either are simple which goes to coding or it needs specification which goes to specification form Integration.

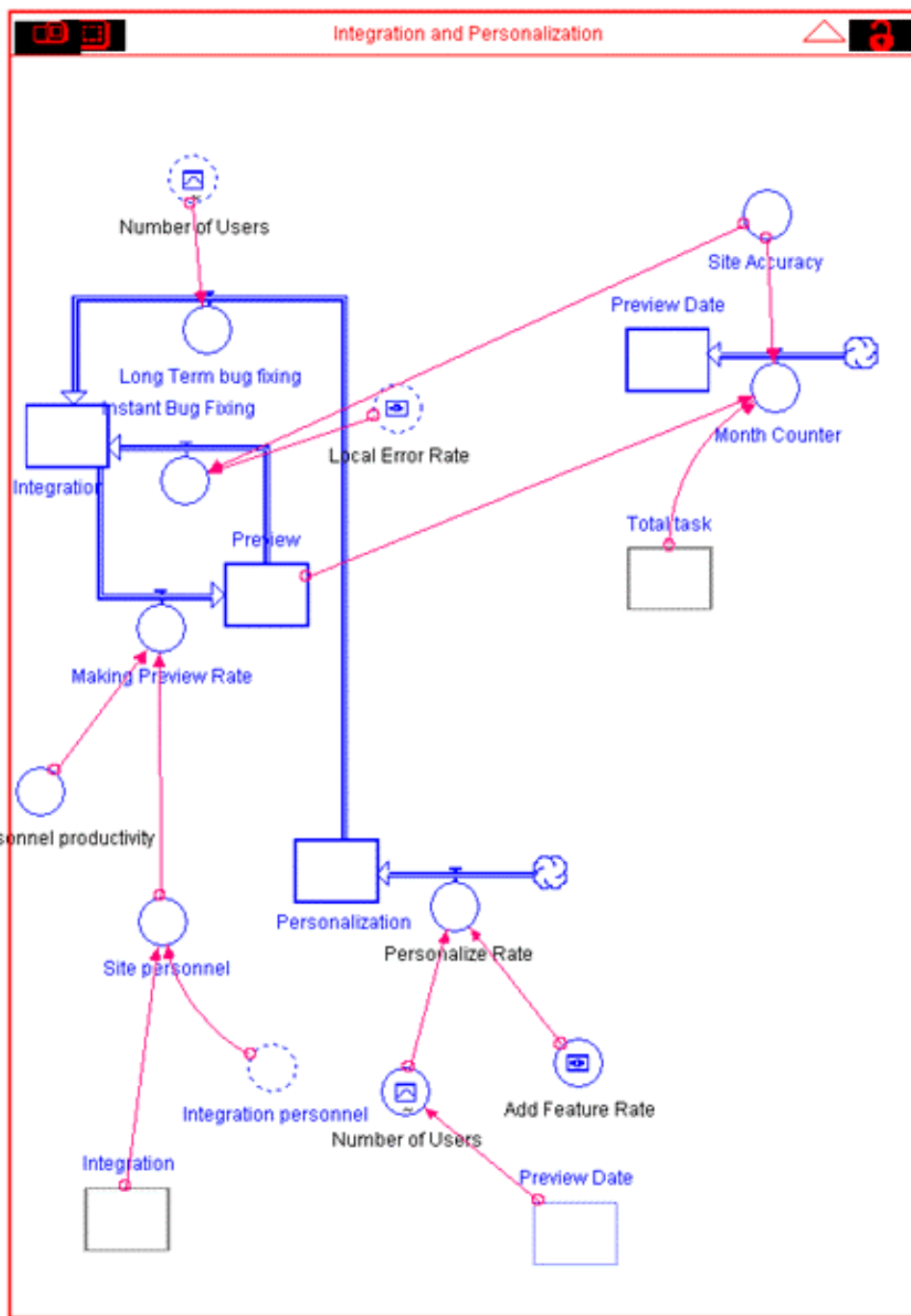
- Coding rework rate: This rates depends to a couple of factors. We believe as the number of number of Integration and Personalization works increase it generates more rework for coding direct and indirect. We show the direct factor here as the rework rate is the sum of rework rate of personalization and Integration plus the local rate.
- Test and QA(function point): This receiver is basically to double-check the coding. This stage is as important as other test stages.
- Integration: Integration is the most important component in our model. The coded module integrate with the outsource.

#### 4.5 Instant Debugging and Site Development

This part is probably the most specific module for IRAD. The integrated software will be ready to put as a preview site. The preview site is a running site with all features. The personalization is also depends on the click rate. This module is consist of the following components:

- Preview(function point): This is the main result of the IRAD. A full feature running site. There is a feedback form preview to Integration for possible instant bug fixing when w notice a simple error on the site and we have to fix it quickly.
- Preview Date (time): When the amount of preview, which is ready, is very close to the amount of the work at the begging w assume the site is ready. Preview date shoes the month which site could be put on the net.
- Site accuracy: This is the desired accuracy, which we consider when the difference is less than .1 is an acceptable work. We can change this work as see how system oscillate sometimes as you can never reach the point as you have to add new features to the current site as more users click on the site.
- Personalization(function point) By personalization we mean the code which has to write to make a personal view and account for each user. As the number of users goes up the possibility to rework some of the previous job also goes up. Also it depends how many feature we want to add. There is a feedback from Personalization to coding and to Integration, which covers feature adding and long term bug fixing.
- Number of users: We assume an exponential increase in the number of users. This numbers is a common belief and it depends how successful is a site.

#### 4.6 Human Resource



**Figure 8**

Another crucial factor in Internet companies is human resource strategy. While the number of people should keep as low as possible but as the rate of development is fast an optimal number for the whole project and an optimal number for each part of the project is a challenge. The major factors are:

- Communication overhead: The old models for communication does not work for IRAD. Not only the number of people who communicate is less then the number, which usually considered in traditional model; people communicate in different ways. We mention to these tools in previous section.

- Hiring Rate(person/month): we assume project will hire a new engineer every month, Finding some key engineer takes time but as they are very experienced people it worth.
- Assimilation rate(person/month): We assume about 25% of hired people is very smart and they are high paid personnel so they learn with a faster rate than others do.
- Quitting rate(person/month): Even though there is very rare for personnel to quit in an Internet companies but as the new Internet companies come to market every day there is always a chance for engineer to quit. We consider this rate as a very small rate.
- Development rate(function point/month) development rate basically is calculate based on the effect of communication overhead on the number of developers times the productivity rate. When developers are involve in two modules we get the maximum works in two receivers to make sure enough personnel specify to do the task.

Development rate computed with the same strategy for the rest of the model.

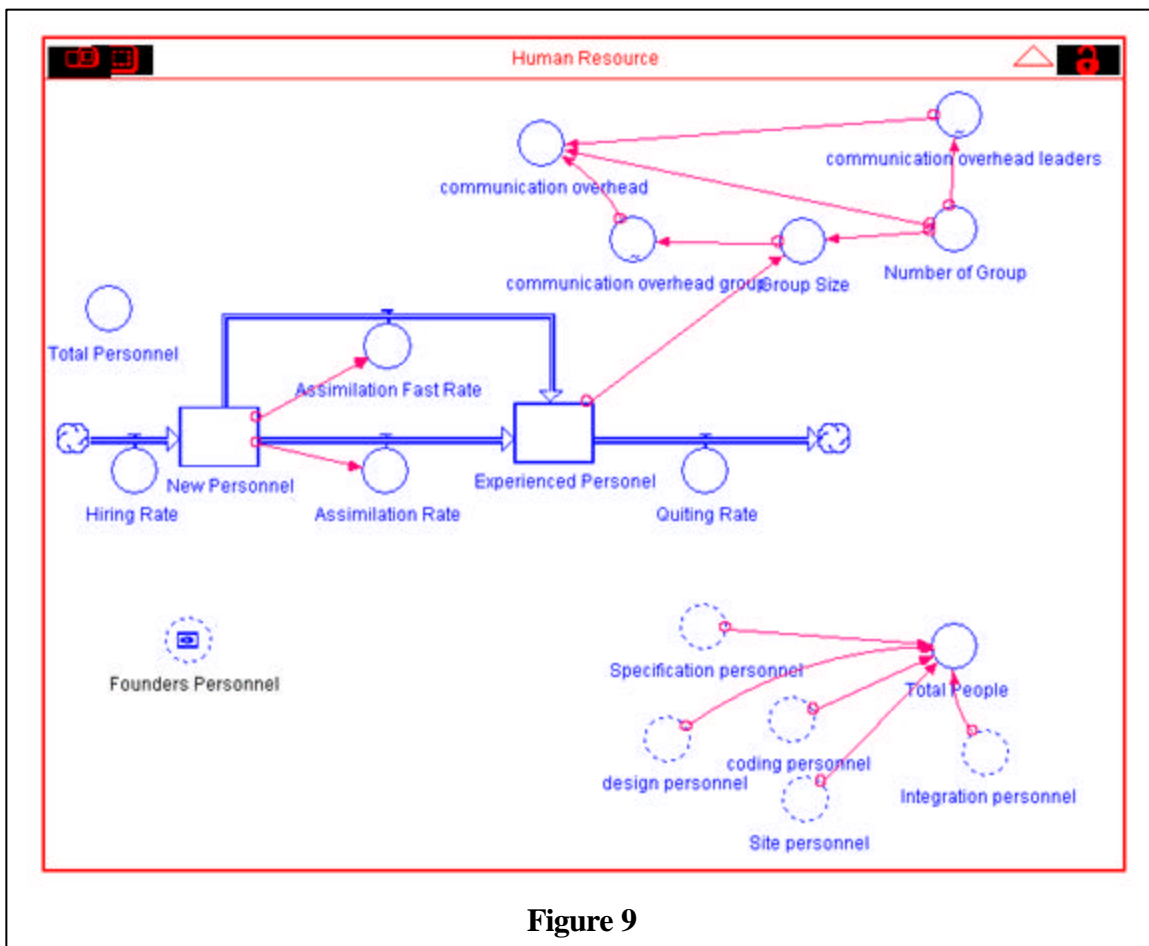


Figure 9

#### 4.6.1 Staffing Strategy

The staffing strategy is very crucial in IRAD. We use of a very simple but very efficient way as our strategy. We consider a proportional model for each part of development. However we make sure there is no inactive personnel in development. For the specification at the beginning we consider only founders spend time on that part. For each stage and relevant test module we consider the same group of the people work on both part.

## 5 Model Verification and Validation

### 5.1 Result

Fig 10 shows the dynamic process of an Internet company. There is couple of issues, which we would like to address in this section.

**These issues are as following:**

- As we see module execution follows in the order of considered time frame.
- This project basically has no end. Personalization, Instant bug fixing and adding features adds more and more work at the end of the process.
- While a preview may be finished when 99% of work is done, some work may add at the end of the site as soon as site goes on the net.
- The dependency among specification, test and design plays an important role in the whole process.
- If outsourcing would be reliable it helps in the process of the project.

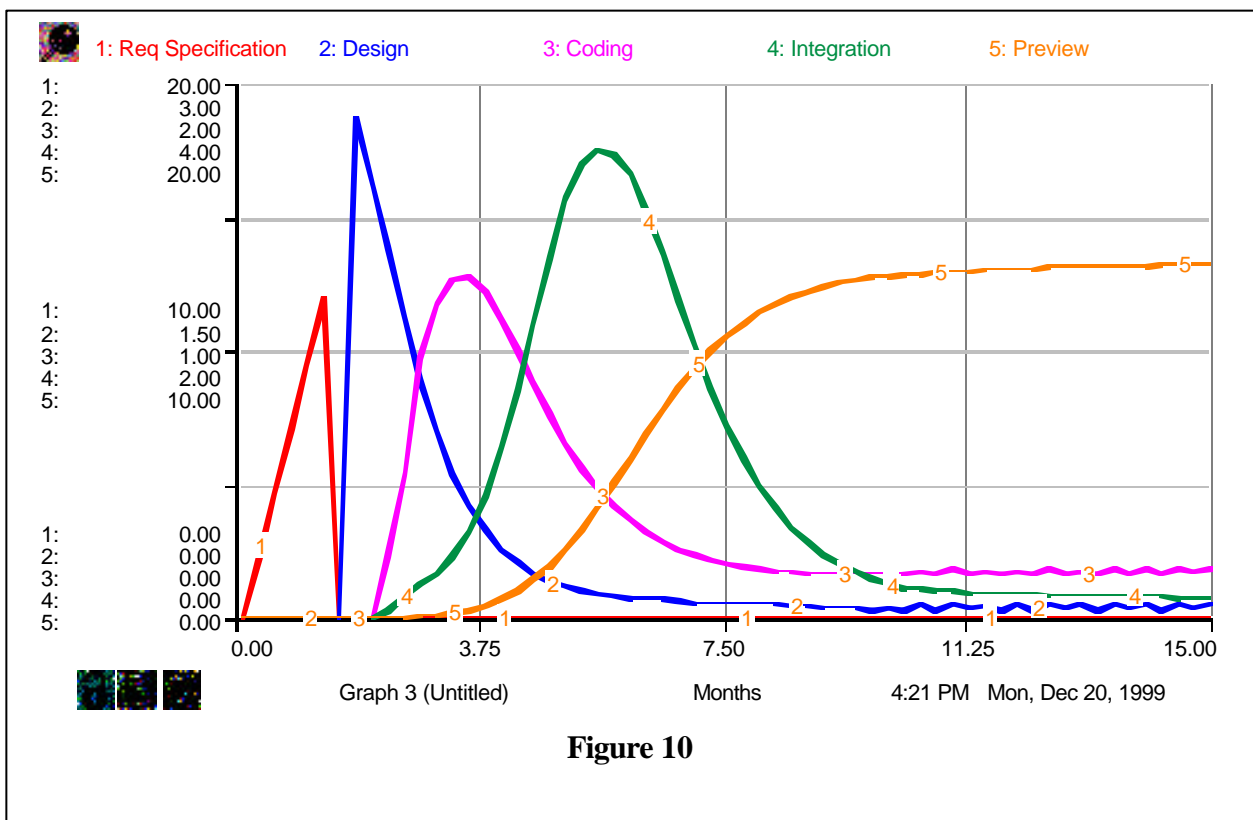
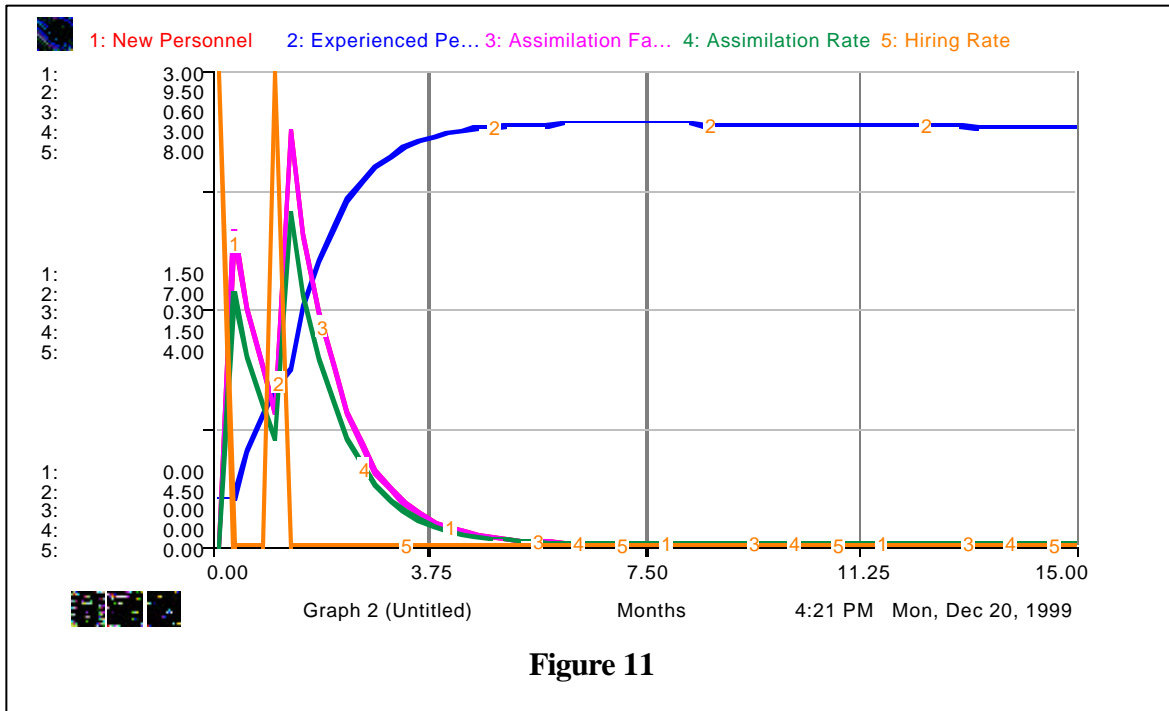


Fig 11 shows the rate of hiring and the rate of hiring new personnel. Basically it takes at least one month to hire a new senior personnel. As the number of Internet companies with good options increase every day it is very time consuming for Internet companies to hire a new engineer. In first few month of the project almost at then end of specification rate the team should be a complete team for execution.



The following table shows the distribution of staff in each module

Month	Specification Requirement	Design	Coding	Integration
Initial	0.00	0.00	0.00	0.00
0	9.57	0.00	0.00	0.00
1	0.00	2.43	0.00	0.00
2	0.00	1.05	1.17	0.30
3	0.00	0.39	1.12	1.20
4	0.00	0.17	0.65	3.06
5	0.00	0.11	0.36	3.23
6	0.00	0.08	0.24	1.96
7	0.00	0.08	0.18	0.91
8	0.00	0.05	0.16	0.40
9	0.00	0.05	0.16	0.20
10	0.00	0.07	0.18	0.15
11	0.00	0.07	0.17	0.13
12	0.00	0.07	0.17	0.11
13	0.00	0.07	0.18	0.10

To illustrate the distribution and relation of hiring strategy we put the numbers in the following table.

<b>Month</b>	<b>Experienced Personnel</b>	<b>New Personnel</b>	<b>Assimilation Fast rate</b>	<b>Assimilation Rate</b>
Initial	0.00	5.00		
0	0.84	6.15	0.23	0.93
1	1.11	7.88	0.35	1.39
2	0.35	8.63	0.15	0.61
3	0.11	8.87	0.05	0.19
4	0.04	8.94	0.02	0.06
5	0.01	8.96	0.00	0.02
6	0.00	8.96	0.00	0.01
7	0.00	8.96	0.00	0.00
8	0.00	8.95	0.00	0.00
9	0.00	8.95	0.00	0.00
10	0.00	8.94	0.00	0.00
11	0.00	8.94	0.00	0.00
12	0.00	8.93	0.00	0.00
13	0.00	8.93	0.00	0.00
14	0.00	8.92	0.00	0.00

**Table 3**

We show the distribution of testing process in different modules in Table 4.

<b>Month</b>	<b>Test and QA- Spec.</b>	<b>Test and QA Design</b>	<b>Test and QA</b>	<b>Design test Rate</b>	<b>Spec. Test Rate</b>	<b>Coding Test Rate</b>
Initial	0.00	0.00	0.00			
0	0.00	0.00	0.00	0.00	0.00	0.00
1	8.75	2.82	0.00	2.82	14.15	0.00
2	2.89	6.92	1.56	7.54	0.30	1.74
3	0.87	4.44	5.07	2.98	0.30	4.95
4	0.26	2.04	5.15	1.14	0.30	3.77
5	0.07	0.82	3.36	0.55	0.30	2.12
6	0.00	0.34	1.92	0.40	0.06	1.22
7	0.00	0.11	1.17	0.30	0.00	0.85
8	0.00	0.06	0.82	0.27	0.00	0.69
9	0.00	0.04	0.65	0.18	0.00	0.61
10	0.00	0.06	0.45	0.19	0.00	0.53
11	0.00	0.05	0.37	0.22	0.00	0.61
12	0.00	0.05	0.31	0.23	0.00	0.60
13	0.00	0.05	0.27	0.23	0.00	0.60
14	0.00	0.05	0.24	0.23	0.00	0.61

**Table 4**

The rate of the teamwork is also important. The following table shows how the rate of the work changes during the time. The interesting point is that Integration is a main factor in the whole process. Also it is interesting to see how the rate of coding will be stable at the end of the process

<b>Month</b>	<b>Specification Rate</b>	<b>Design Rate</b>	<b>Coding Rate</b>	<b>Integration Rate</b>
Initial				
0	9.57	0.00	0.00	0.00
1	4.43	5.25	0.00	0.00
2	0.00	5.85	2.62	0.23
3	0.00	2.02	4.30	0.60
4	0.00	0.62	2.70	0.60
5	0.00	0.19	1.22	0.60
6	0.00	0.07	0.49	0.60
7	0.00	0.00	0.19	0.60
8	0.00	0.00	0.07	0.60
9	0.00	0.00	0.05	0.59
10	0.00	0.00	0.07	0.43
11	0.00	0.00	0.06	0.34
12	0.00	0.00	0.07	0.28
13	0.00	0.00	0.07	0.24
14	0.00	0.00	0.07	0.22

**Table 5**

If we add the rate of the work in each month this what we get as the effort for every month. Fig. 12 shows this distribution.

We may have also the percentage of the work in each module

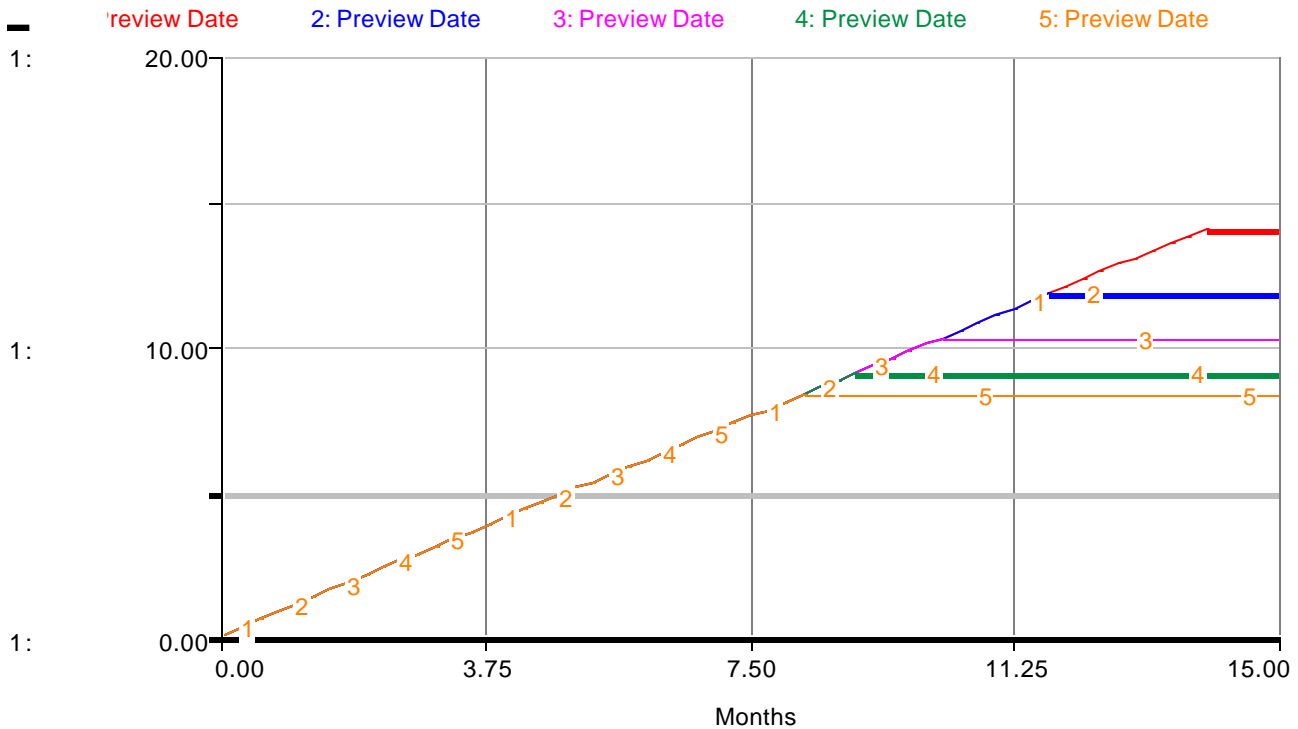
## 5.2 Model Sensitivity[7]

In this section we test the sensibility of our model to the parameters; at first we try to test the sensibility to the preview date as this is a very crucial factor for the development.

1- The effect of change in “Initial experienced personnel” on “Preview Date”

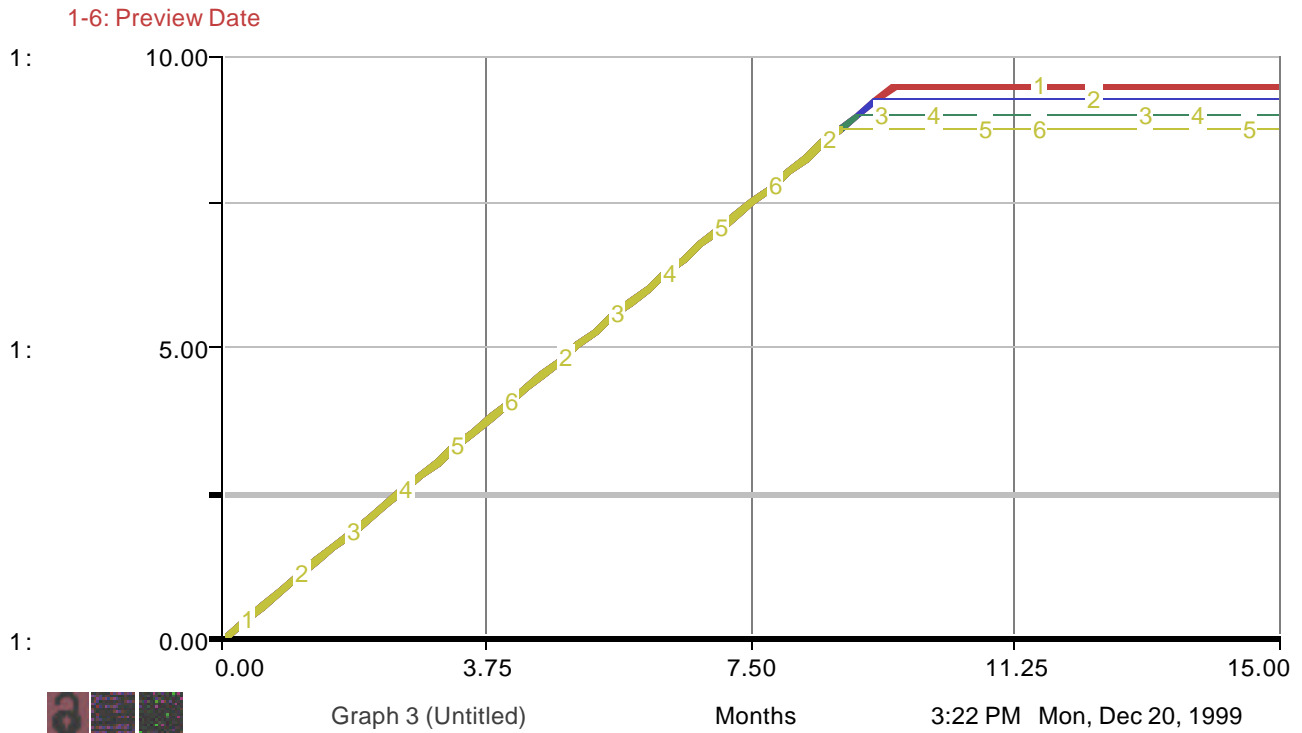
	Run #1	Run #2	Run #3	Run #4	Run #5
Number of Initial experienced personnel	2	3	4	5	6

As we see in this picture the number of founders has a crucial role in development duration.



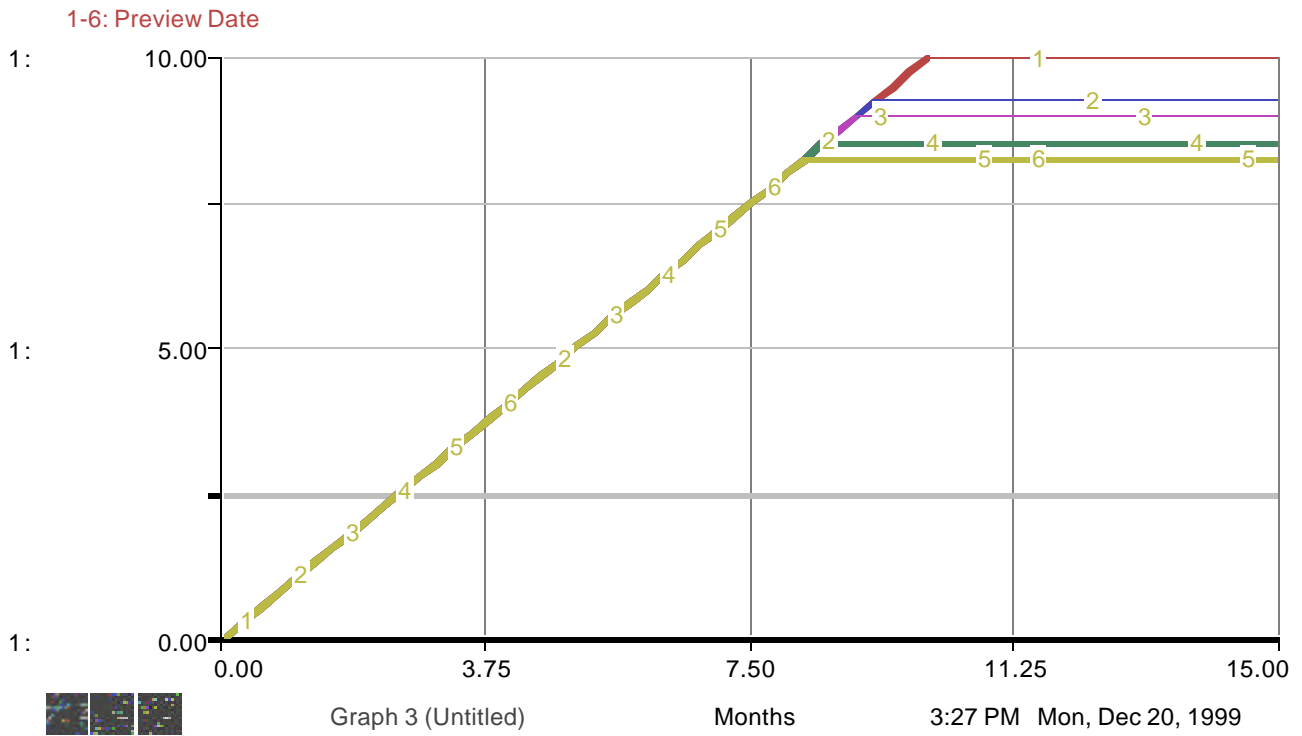
2- The effect of change in “Specification Personnel Productivity” on “Preview Date”

	Run #1	Run #2	Run #3	Run #4	Run #5	Run #6
Specification Personnel Productivity (Function points / person-month)	1	1.2	1.4	1.6	1.8	2



3- The effect of change in “Design Personnel Productivity” on “Preview Date”

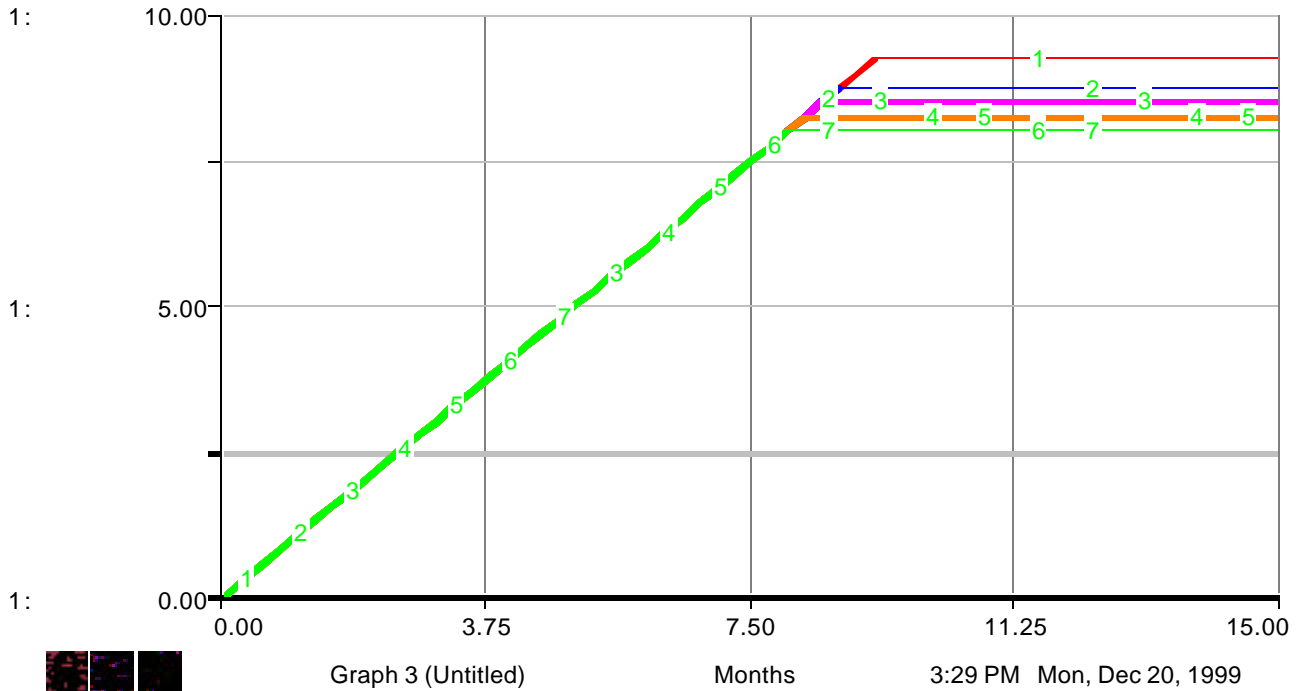
	Run #1	Run #2	Run #3	Run #4	Run #5	Run #6
Design Personnel Productivity (Function points / person-month)	1	1.2	1.4	1.6	1.8	2



4- The effect of change in “Coding Personnel Productivity” on “Preview Date”

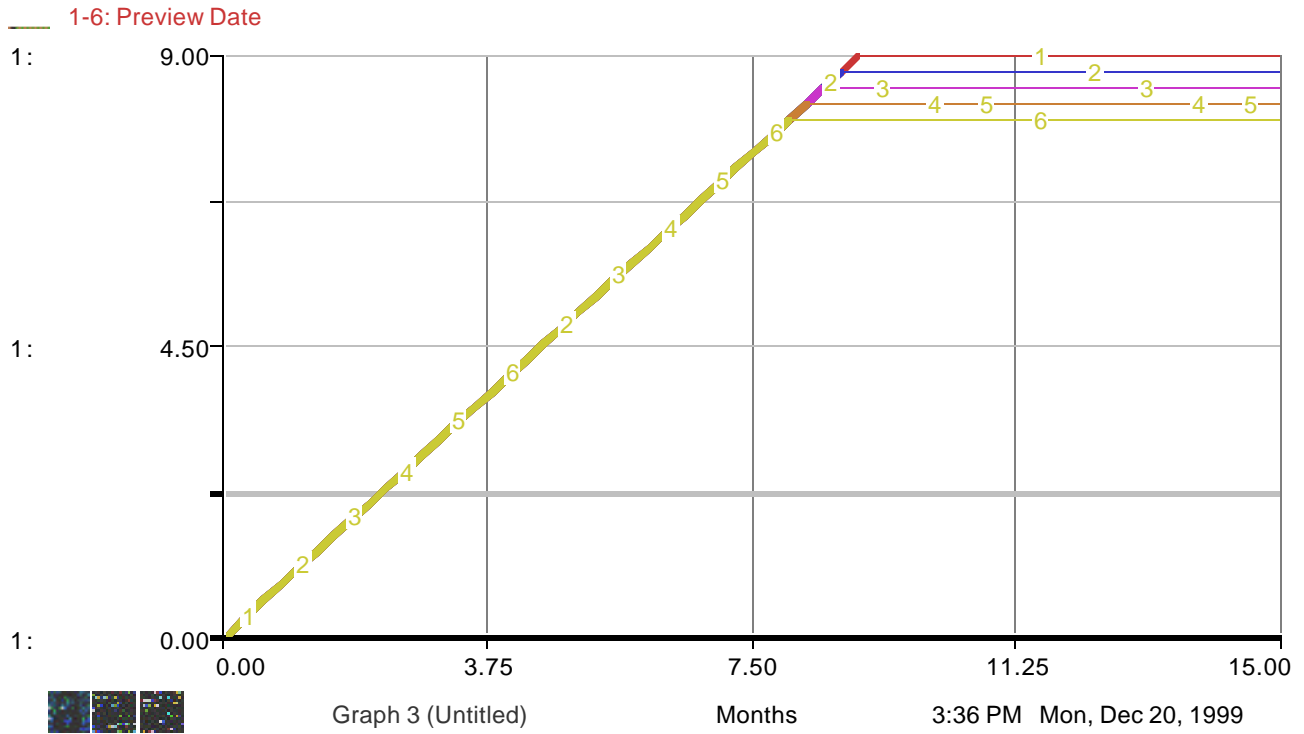
	Run #1	Run #2	Run #3	Run #4	Run #5	Run #6	Run #7
Coding Personnel Productivity (Function points / person-month)	0.8	0.9	1.0	1.1	1.2	1.3	1.4

1-7: Preview Date



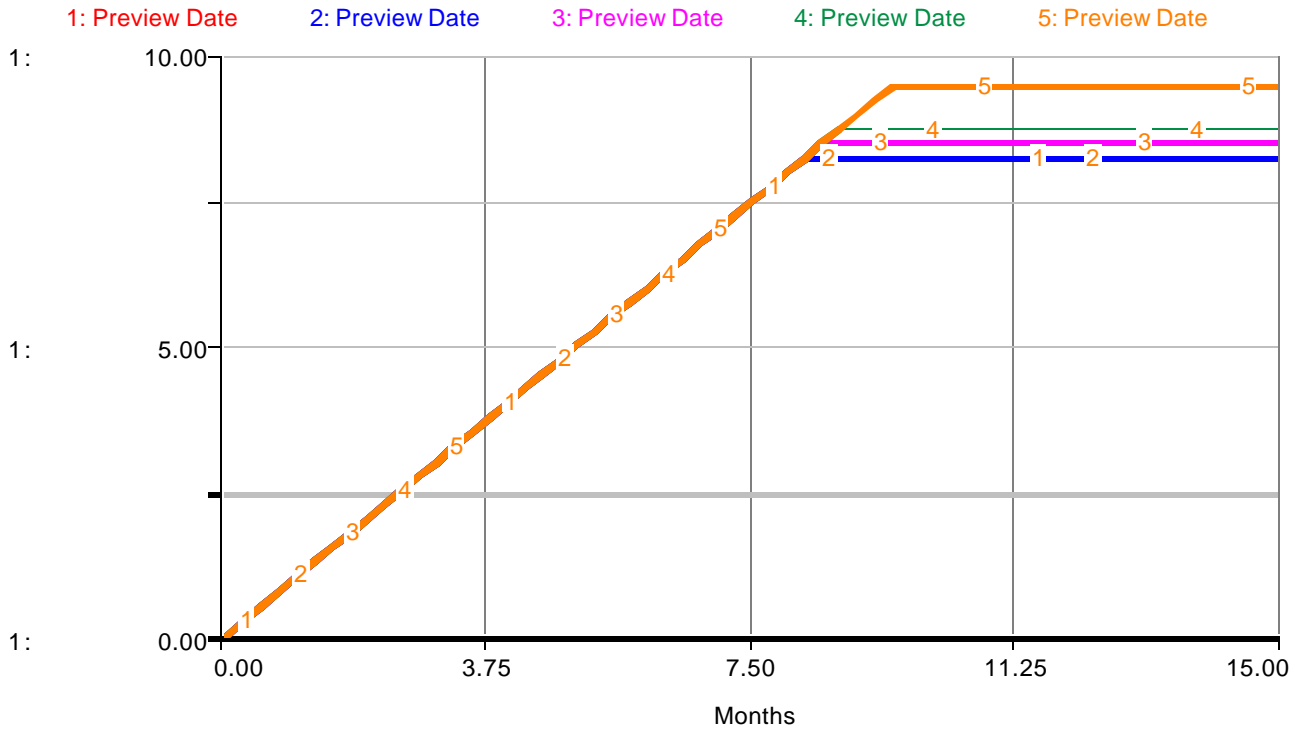
5- The effect of change in “Integration Personnel Productivity” on “Preview Date”

	Run #1	Run #2	Run #3	Run #4	Run #5	Run #6
Integration Personnel Productivity (Function points / person-month)	0.8	0.9	1.0	1.1	1.2	1.3



6- The effect of change in “Local error rate” on “Preview Date”

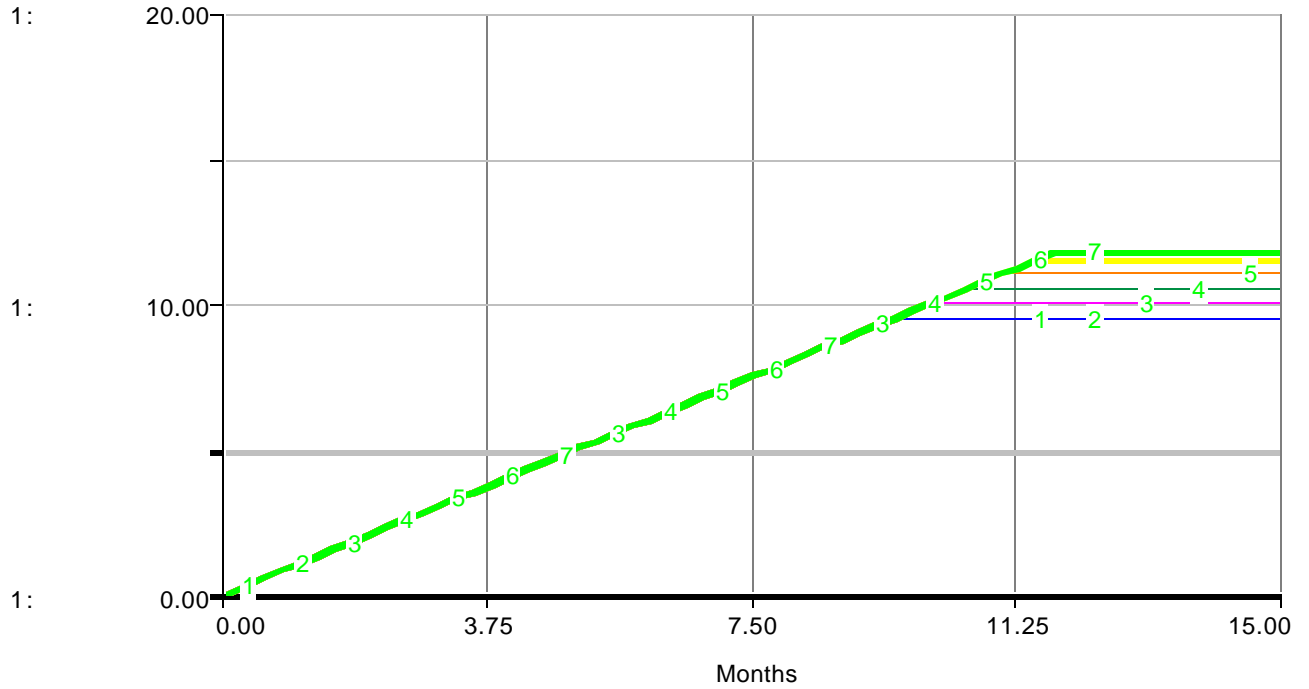
	Run #1	Run #2	Run #3	Run #4	Run #5
Local Error rate (defect/month)	0	.05	.1	.2	.3



7- The effect of change in “Integration Error Rate” on “Preview Date”

	Run #1	Run #2	Run #3	Run #4	Run #5	Run #6	Run #7
Integration Error rate	0.001	0.010	.030	0.050	0.070	0.090	0.1

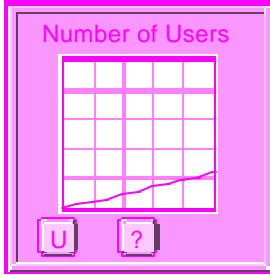
1-7: Preview Date



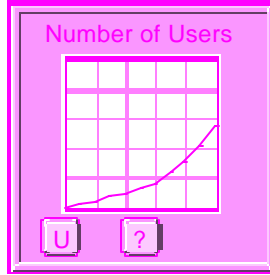


9- The effect of change in “Number of Users” on “Coding”

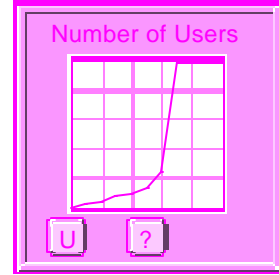
Run #1



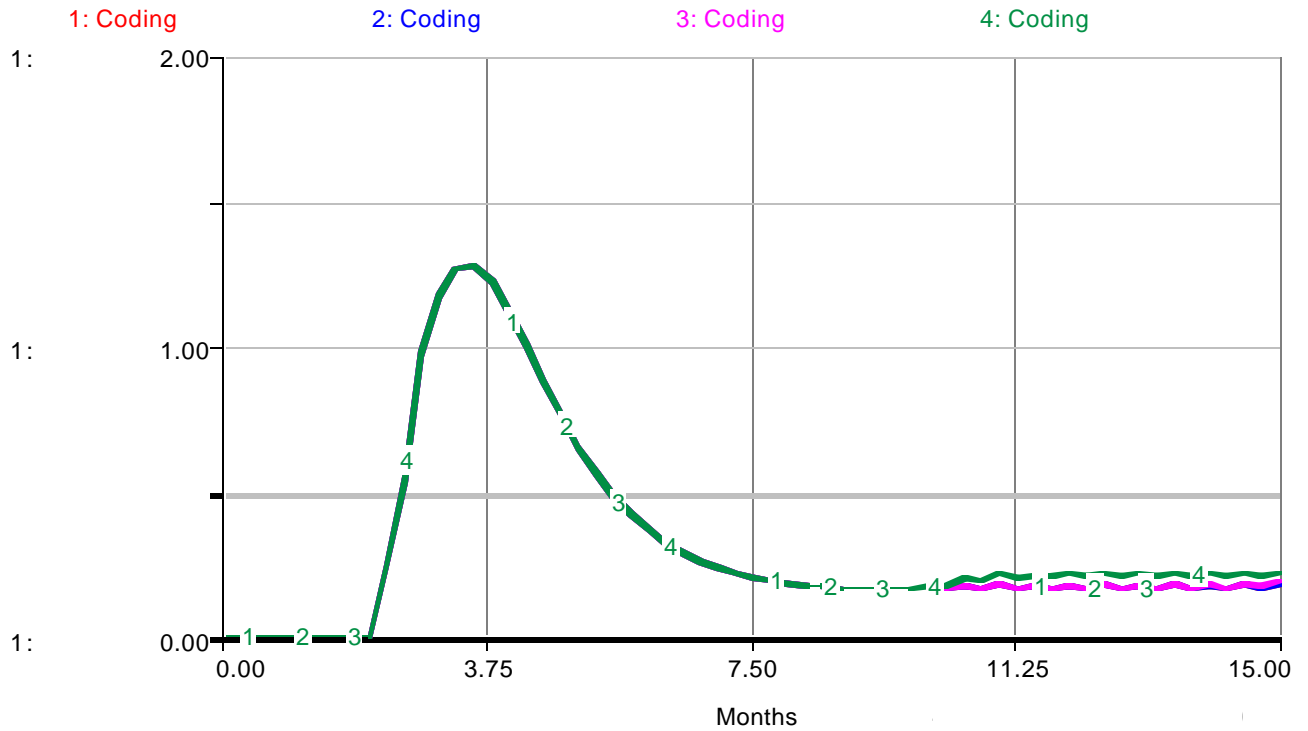
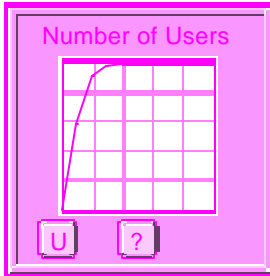
Run #2



Run #3

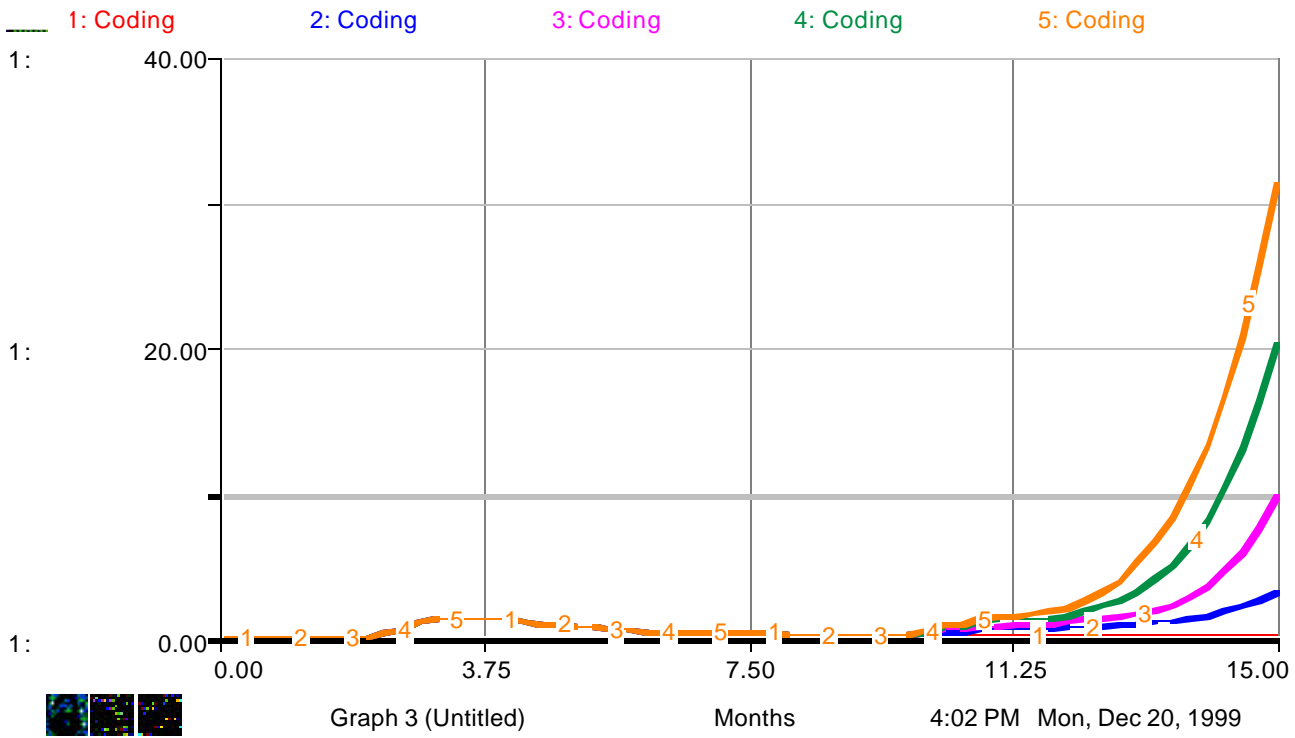


Run #4



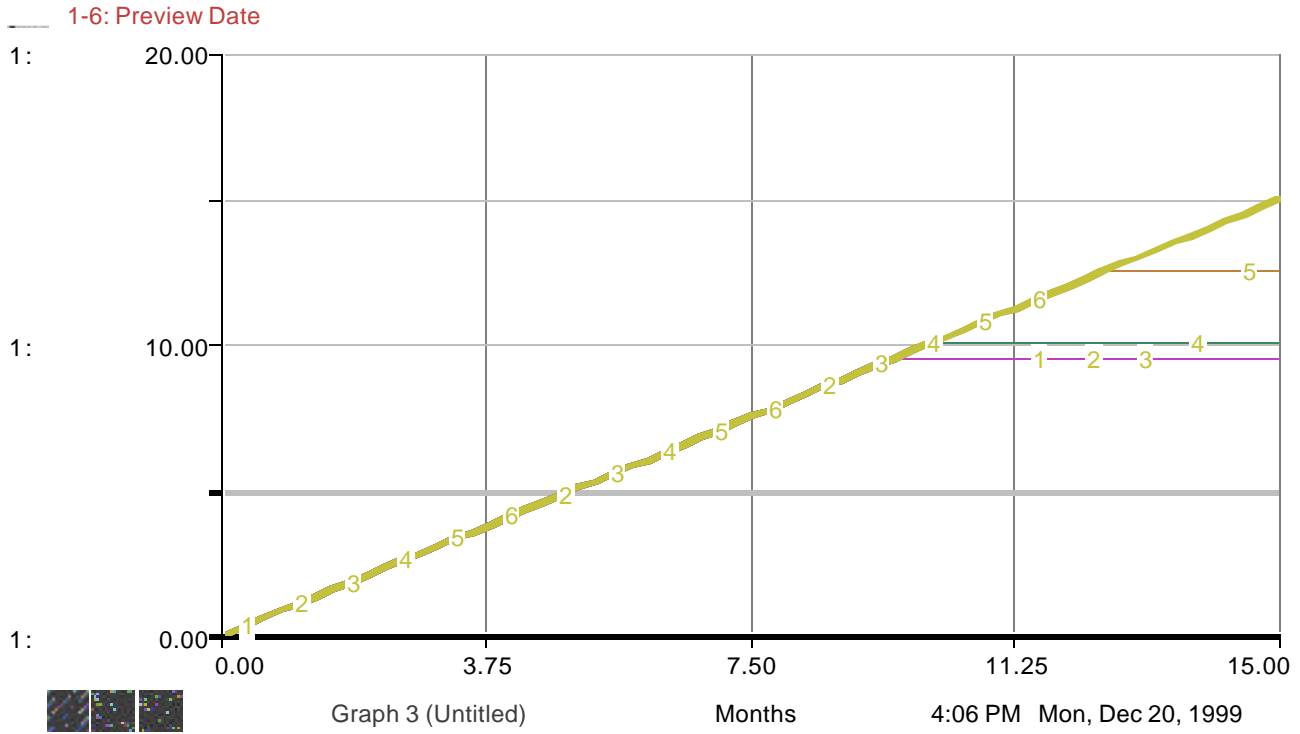
10- The effect of change in “Add feature Rate” on “Coding”

	Run #1	Run #2	Run #3	Run #4	Run #5
Add feature rate	-6	-6	-6	-6	-
	1*10	250*10	500*10	750*10	6
					1000*1
					0



11- The effect of change in “Outsourcing Percentage” on “Preview Date”

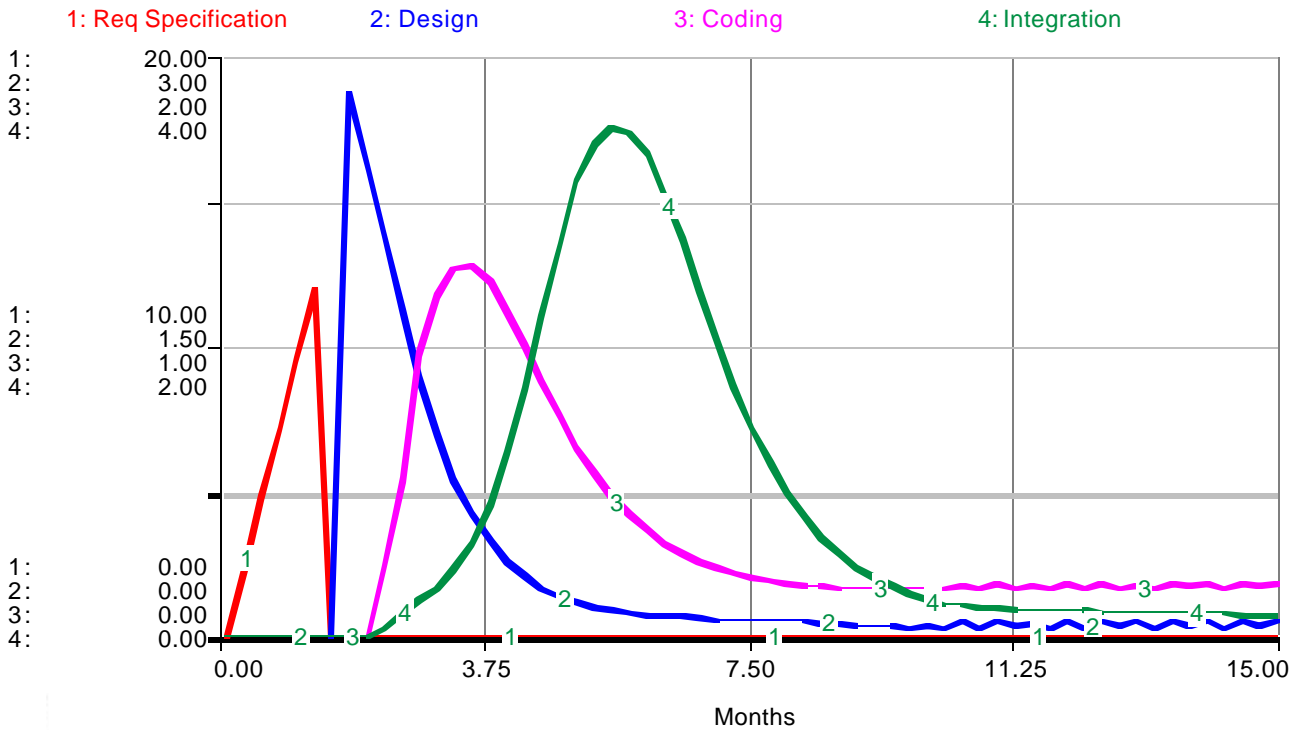
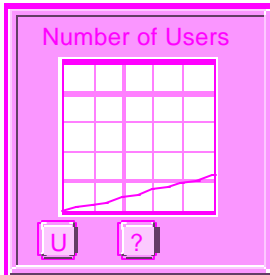
	Run #1	Run #2	Run #3	Run #4	Run #5	Run #6
Percentage of outsourcing	0	5	10	50	75	100



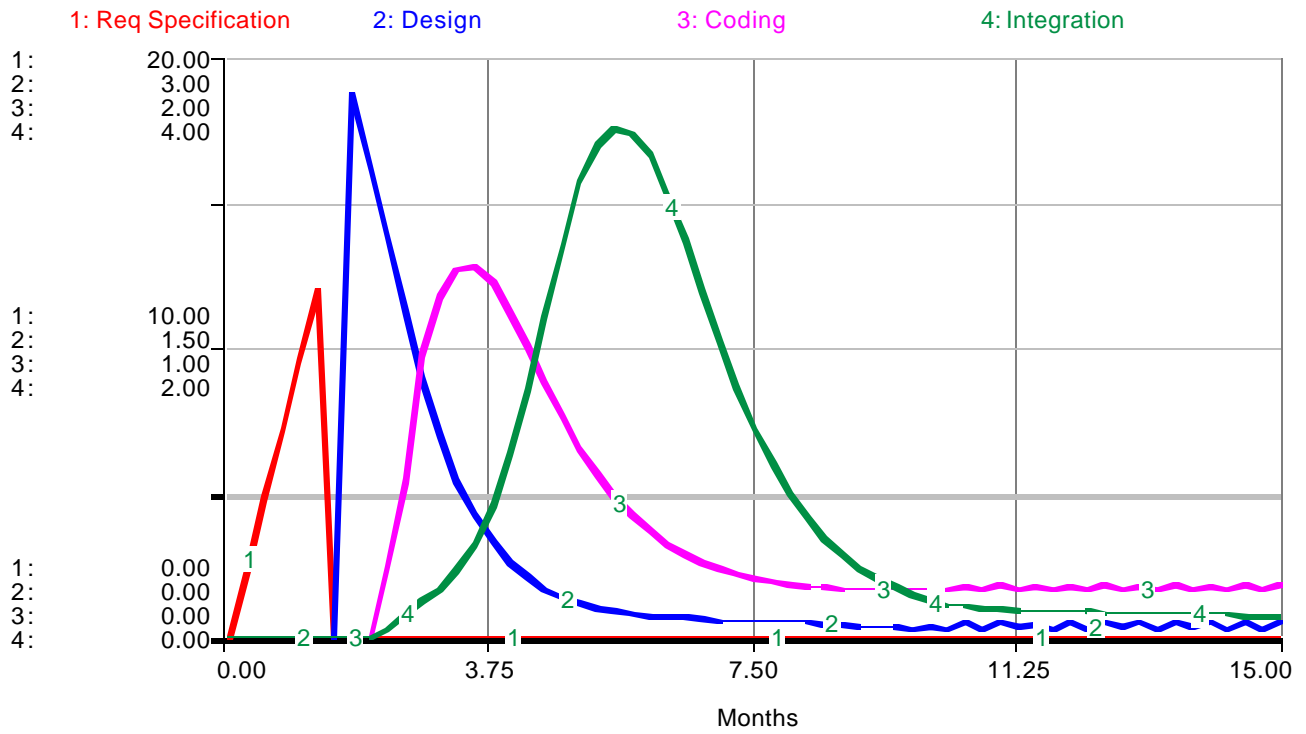
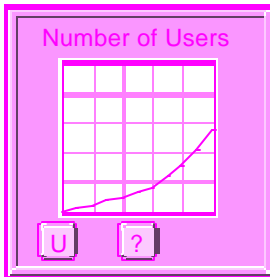
## 12- The effect of change in “Number of Users ” on “Specification “, “Design” and “Coding”

The Internet user increase exponentially, if it happens to a site the main effects goes to coding and Integration. As the number of users goes high a site has to attract more users so they may add new features every quarter or even every month. The rate of success is a positive feedback as you get more successful you get more attention so on.

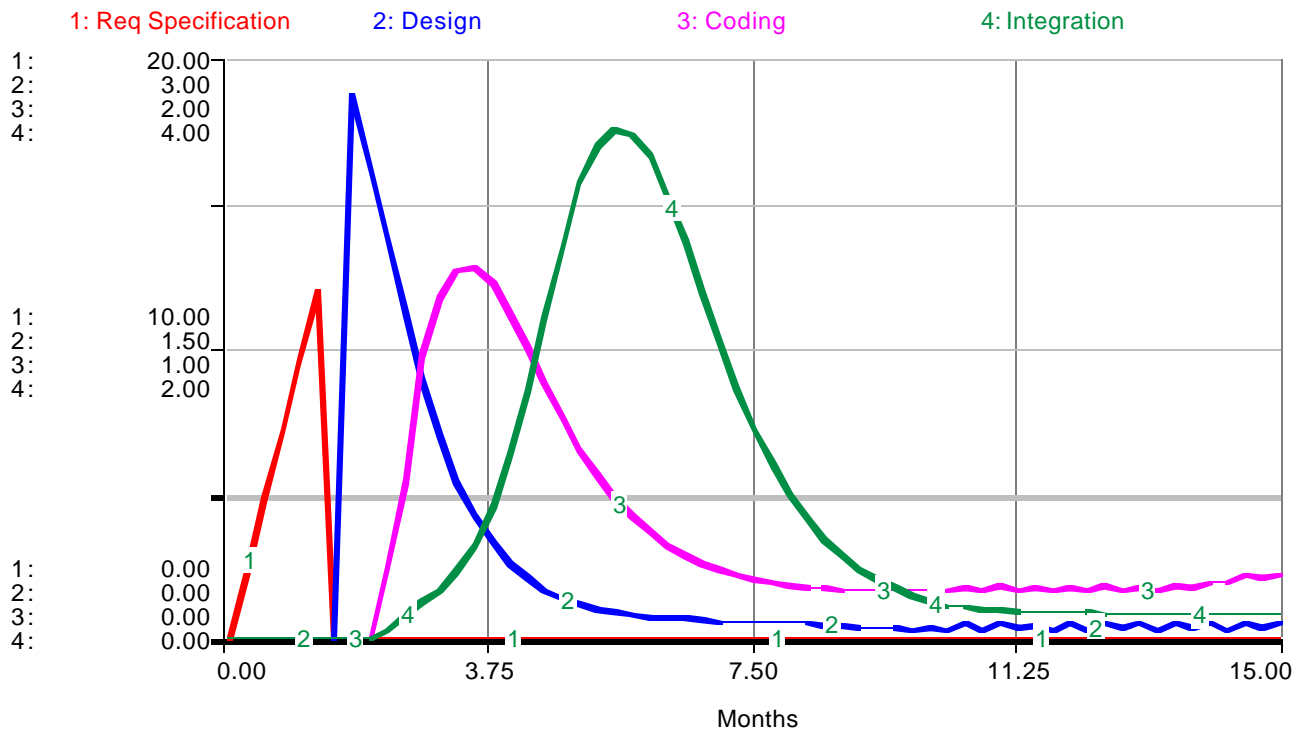
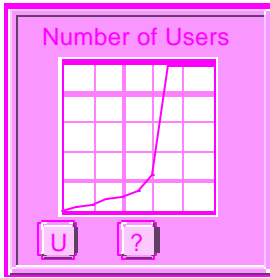
Run #1



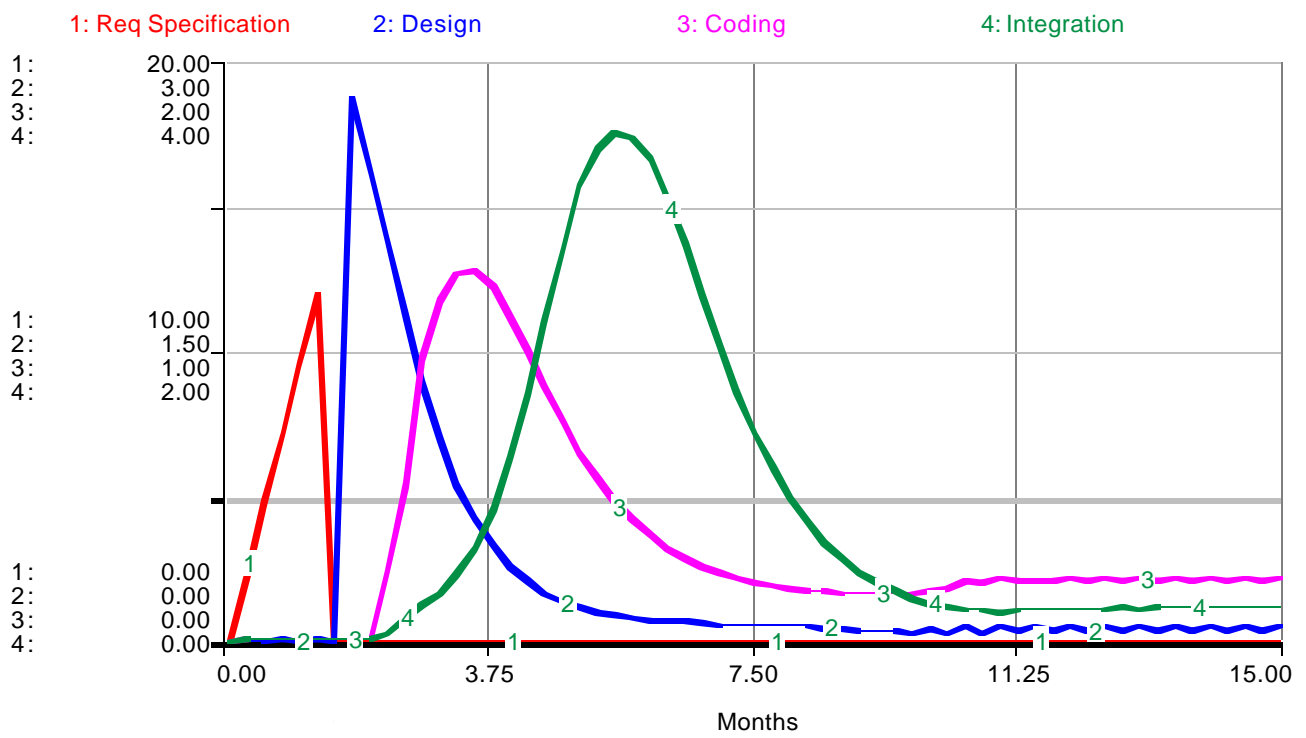
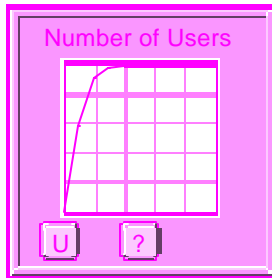
# Run #2



# Run #3



# Run #4



### 5.3 comparison to references behavior

Considering our assumptions the model follows the reference behavior. However as the human resource model do not watch the total resource we just double check to make sure the total personnel match with what the project needs.

### 5.4 Expert review

To evaluate our model we interview Dr. Amir Zarkesh the CEO and co-founder of u4cast.com. u4cast is a startup Internet Company with emphasis on forecasting and data mining. u4cast.com is in the first round financing and they started a prototype (not a Preview) for the sake of fund raising and to show the concept.

We found out as we thought the main concern of u4cast is the explosion of users in the first few month so some issues as load balancing and site crash is an important issue. The estimated site development for u4cast is about 9 month while they have a fall back scenario in case if they could not get the money they are looking for.

## 6 Model Application and Transition

### 6.1 Result Analysis

One of the major issue in terms of prediction of the work is the percentage of the working each module. In our questionnaire we asked:

What is the approximate percentage of: *Specification, Design, Development, Test and Maintenance*?

The following chart shows the distribution of four major part and the relative ratio to each other. As we expected Integration and specification are the main concern in IRAD. Our simulation shows this characteristics. This is also compatible with our findings from CTO's and CEO's of interviewed companies.

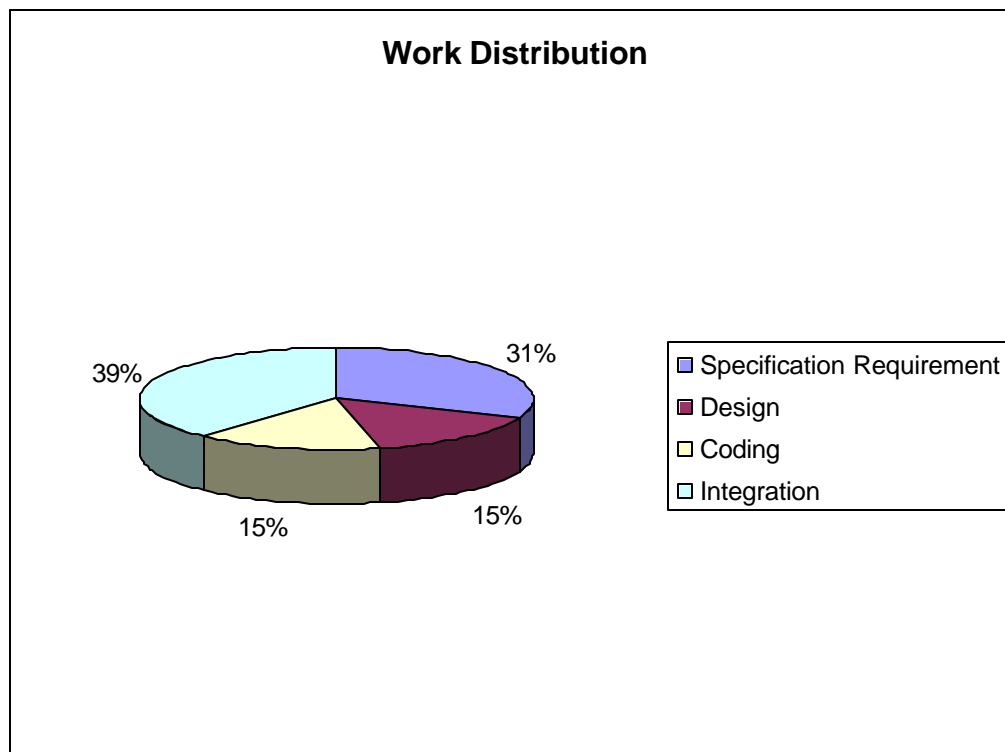


Figure 12

Fig. 13 shows the distribution of effort during the whole project. As it's expected after first 9 month there is less work to do. We have to mention we hadn't shown the rate of preview work in this figure.

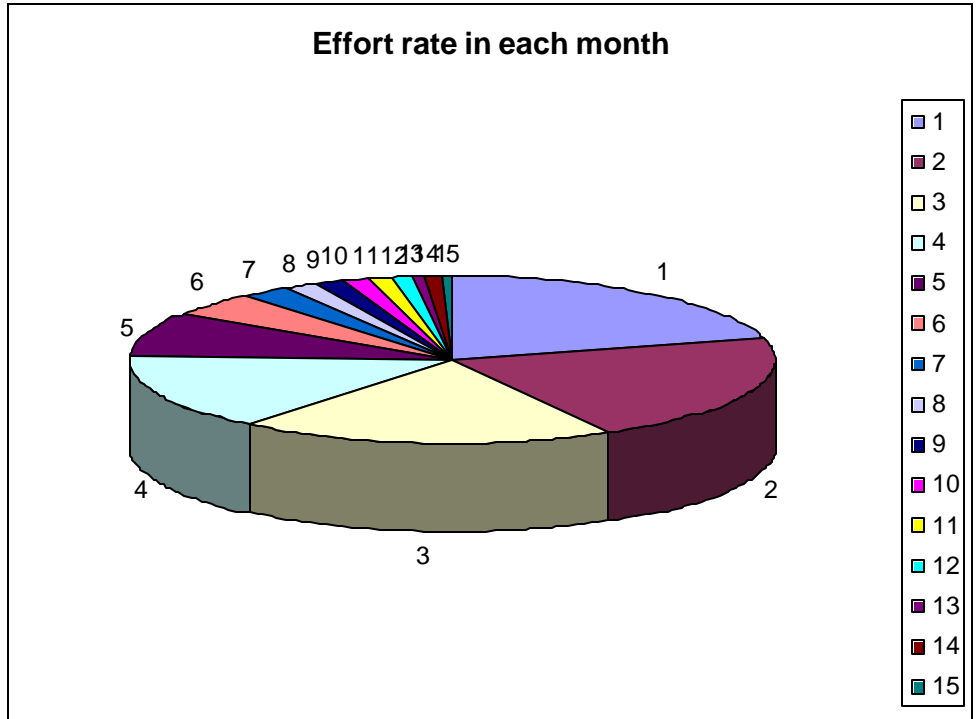


Figure 13

## 6.2 Limitation of the model and future enhancements

In current model we've just simulated a process of developing a site from beginning up to now. Although we considered the effect of external factors like market pressure implicitly, we didn't simulate such factors directly.

Currently the model doesn't show the detail process of test and QA. We may want to expand the test and QA parts as a more detailed sub-model in our model in future.

Outsourcing was not showed as a complete and expanded sub-model in our model. We just tried to show the process of coding and implementation in Internet companies. With the aid of current model we could measure the amount of outsourced work and the system sensitivity to different amount of outsourcing.

## 6.3 Next Steps

To improve our model and make it more efficient, we are going to add following parts to the model which we believe are really effective to shorten the schedule duration:

- Tuning
- Inspection
- Feature clustering

### 6.3.1 Tuning

We would like to tune more our assumptions with the more realistic data. To achieve such a goal we are still in negotiation with several Internet companies to give us more detail about their work. Even though this work may takes time but we believe even in short run this model evolves to an accurate model.

### 6.3.2 Inspection

Inspection is beyond this paper we would like to apply the model designed by Ray Madachy as part of this model for future. Fore more information about Inspection please look at \*\*\*

### 6.3.3 Feature Clustering

Internet companies offer a lot of features with their service to their customers. These features not only change month by month but also the number of features provided also increase. As the competition with other competitors is very important in current market, Internet Companies should be very accurate to select the feature they want to provide and

to make sure they do not miss any important feature. We introduce the idea of feature clustering in which software engineer first find some clusters with the features space of its products. Within a collection of objects a cluster could be defined as a group of objects whose members are more similar to each other than to the members of any other group. In each class there are some feature, which are very similar to each other.

Clustering is rather a hard task specially when the number of attributes of objects increases, i.e. when a company wants to add several new features to a running web site they have to consider several factors for each attribute. These attributes could be the cost of the feature, the time of development, the time of specification, time for bug fixing, user feedback about such feature, competitors situation, possible way for outsourcing such feature etc.... When the number of attributes increase a smart way of clustering features would decrease the total cost and increase the efficiency, accuracy and satisfiability of a running site.

In RAD clustering is used to segment a group of features collection into subsets, the clusters, with the members of each cluster being similar with respect to certain interesting attributes. For clustering no predefined taxonomy or classification schemes are necessary.

Typically, the goal of cluster analysis is to determine a set of clusters, or a clustering, in which the inter-cluster similarity is minimized and intra-cluster similarity is maximized. In general, there is no unique or best solution to this problem. A number of different algorithms have been proposed those are more or less appropriate for different data collections and interests.

Furthermore, we will try to collect more real data which help us to make our model more robust.

## **7 Conclusion and recommendations**

While Internet business try to keep their IP as the most valuable property inside their company, revealing the secret of Internet companies is still a challenge. This work is one step forward to model the dynamic process of such companies.

Our result from surveys shows how model could be very dependent to the nature of the project. Couple of CTO's indicates that every question in our survey depends on their project. However they believe an overall model would help to find pitfalls in the process of the software.

We did show how outsourcing could take care of the load balancing and site issues even if it is less than 20%. Also we show how in an second-generation startup people who specified task they have to code themselves and in the integration module almost everybody gets involve.

Our finding shows an early error detection may save a lot of money. If a site crash once the cost of having a crashed site is about \$10,000 per minute. However the salary of an experienced engineer could be the same amount for one month.

## 8 References

1. Mc Connel, S., *Rapid Development*. 1996: Microsoft Press.
2. Boehm, B., *Making RAD Work for Your Project*. IEEE computer, 1999(March).
3. [WWW.Outsourcing.com](http://WWW.Outsourcing.com), *Survey of Current and Potential Outsourcing End-Users*. 1998: The Outsourcing Institute Membership, 1998.
4. Roheling, C., Hermann, and Smith-Daniels, *System Dynamics Modeling Applied to Software Outsourcing Decision Support*, . 1999, Department of Computer Science and Engineering - College of Engineering and Applied Sciences, Arizona State University.
5. Lowell, J.A., *Rapid Evolutionary Development*. 1991: John Wily & Sons.
6. Cusumano, M.A., Yoffie, David, *Software Development on Internet Time*. IEEE Computer journal, 1999(October).
7. Forrester, J.W., *An Introduction to Sensitivity Analysis*, . 1996, MIT: Boston.

## Appendix A

The following is a sample of questioner, which we made and sent to several Internet companies.

### **RAD issues for Internet Companies**

Please answer the following question for your company. IF in some question there is a question about a Project please specify the nature of the project.

#### **General Questions**

1. What is the software product category? (The type of business ) *Portal , Sell a product, Offer a service, Offer an Internet/Intranet Service, Others*
2. Type of your Internet company: *B2C, B2B, Communities, etc.*
3. What is your preferred Language for: *Back end, Front end, Web Application Development*

#### **Project Definition**

1. How do you estimate the size of your project? (*Function Point, KLOC, etc...*)
2. How much is the difference between your initial estimation and the real time of project duration in average for an Internet Project?
3. How does market pressure changes your plan? (*Early release, Hire superstars, Outsourcing.*)
4. Do you prioritize features for implementation? How do you classify features (Do you use any specific clustering algorithm?)

#### **Development and Specification**

1. What is the approximate percentage of: *Specification, Design, Development, Test and Maintenance?*
2. Which model or life cycle used in your project development?
3. What percentage of specifications changes during the development? (average)
4. What is the approximate percentage of main differences between the prototype version and production version of your site in the following terms : *GUI, Robustness, Features*
5. Is the production version written almost from scratch or it is mostly improvement of the prototype (i.e. preview) version?
4. What is the approximate percentage of off-shelf solutions and the in-house developments?
6. What is the approximate percentage of effect of off the shelf tools in development
7. How do you do source management and version control: *Sourcesafe, Cvs, RCS? Other?*
8. What is your policy for documentation control?

## **Human Resources and Policies**

1. What was the initial size of your company?
2. Do you prefer to hire **senior expert** personnel (much more cost) to meet your schedule on time or hire **expert** personnel to reduce the cost of project?
3. What is the typical learning curve for new personnel?
4. Do you follow Rayleigh model at the time of hiring project personnel?
5. If a key person leaves the project in the middle of the work, which strategy works better? Hiring new personnel with the same level of expertise or substitute one the old member of project?
6. What is the approximate percentage of programmers in terms of founders /old-timers vs. new hires?
7. What is the typical group structure: *Leaders/senior programmers, Junior programmers, Tech writers, Testers*

## **Communication**

1. Which policy is used in your project to reduce the communication overhead between project members? Electronic mail, Electronic bulletin board, etc.
2. What is your proffered optimal size for smaller groups in your company?

## **Outsourcing**

1. Which projects have you outsource? (what percentage)
2. How successful has been the outsourced project respect to in-house deployments?
3. How do you create an interface between your internal group and outsource company?
4. What kind of risk management policy do you consider for outsourcing?

## **Defect detection policies and Quality Assurance**

1. What is the percentage of instant bug fixing vs. phase bug fixing?
2. What is the mechanism for the bug reporting from the customers to developers?

## **Reusability and documentation**

1. What percentage of your code is reusable? Is it more important to get the software out on the deadline, or the long-term reusability issues are important to consider even against the deadlines.
  2. What is your code documentation policy? Any policy in place or the goal is to make the deadline happen and no room for fancy documentation?
  3. How much is the relative importance of the following features in your product:  
*Portability, Scalability, Speed, Fault Tolerance, Robustness*
- Which tools and policies are they standardizing on for software development (*gcc? VC++, Jbuilder....*)