Lines of Innovation in Software Testing

Stuart Reid

When you think of 'innovation', do you conjure up images of a mad professor, who emerges from his laboratory to find the world has passed him by? Perhaps you think of large organizations investing to ensure they leapfrog their business rivals. Or, do you think of yourself undertaking leading edge challenges that could help the wider community? Well, if you chose the first option, you probably want to read an article on 19th century inventors, whereas if you chose either (or both) of the other options then you are in the right place.

Introduction

This paper will cover some of the concepts behind innovation and explain why businesses must take it seriously, especially during a recession. We will introduce the idea of 'lines of innovation' in software testing to show how combinations of previous innovations can lead to new innovations and see how this has worked in the past and may work in the future. We will also invite the testing community to contribute to improving and expanding this model of our industry. Finally we will consider how we can best encourage innovation, from both the top-down view of management and from the bottom-up view of the tester who would like to see themselves as an innovator.

> ***"You have to do a huge amount of testing, so your upfront costs are greater than ever before. But it's the bit I enjoy. It's exciting and fun."***
> ***James Dyson***

Innovation and Invention

Innovation and invention are closely connected, but therefore often confused. There are various definitions of innovation, but here we shall be inclusive and consider it to be the 'practical use of inventions and discoveries'. Probably the main reason they are often confused is due to the expectation that innovation mainly exploits new inventions, and that 'good' inventions simply become innovations overnight; thus there is often perceived to be little or no gap between invention and innovation. We will see that many innovations are actually based on ideas that have been 'waiting around' for sometimes hundreds of years before a catalyst provides them with their opportunity to shine in a new context. It is also worth noting that it is rare for new inventions to become practically useful in a short timescale (although there are some exceptions) and for many inventions there is a considerable lag (typically 10 years in the 20th century, outside of the world war years) before the invention finds its way into common usage. Perhaps unsurprisingly, testing normally plays a major part in filling this period between invention and successful innovation.

Innovation covers more than just products

Using an all-encompassing view of innovation means that we do not have to limit ourselves to only products. If asked, many can list a number of innovative products that have appeared over the last 100 years or so, such as the light bulb, phonograph, radio, television, and, in the UK at least, the Dyson vacuum cleaner. It should be recognised, however, that innovation does not have to be restricted to products, and in the software testing field we would be severely limited if it was. We can also usefully consider process innovations (e.g. test-driven development), organizational innovations (e.g. TQM) and service innovations (e.g. offshoring). In fact, the majority of innovations from the last century are probably not product-based.

> ***"We haven't been delivering innovation fast enough"***
> ***Stephen Elop, CEO Nokia***

Manufacturer-Based Innovation

Apart from the type of innovation (e.g. product-based, etc.), another way we can classify innovation is based on what drives it. Traditionally, innovation has been thought of as manufacturer-based, with R&D labs seen as the birthplace of new innovations that will each generate the next 20 years of profits. A current example of highly-successful manufacturer-based innovation is provided by Apple with their iPods, iPhones and iPads, whereas Nokia have recently lamented their inability to innovate and cite it as a reason for their loss of market share. This form of innovation is typically dependent on patents, which provide the owner with a 20 year term during which they have exclusive rights to make use of their invention. Patents are considered by some to restrict progress, but the opposing argument is that without patents organizations have no protection for their ideas and therefore less incentive to invest in innovation. By their nature patents are also made public, thus acting as a means of disseminating new ideas and practices. Limiting access to innovations can also sometimes be counter-productive, even when big business is concerned. If you're old enough you may remember Sony's Betamax video recording format that while providing superior picture quality was supposedly overtaken by JVC's VHS format partly due to Sony's restrictive licences on the Betamax technology.

Innovation = Best Business Practice

If done sensibly innovation can become an integral part of a long-term business strategy, aimed at increasing both sales and profits. Of course, during a recession there is the natural tendency to 'batten down the hatches' and concentrate limited resources on optimization of current products and processes. The research, however, does not support this approach of polishing the inexorably ageing apple. In their 2010 survey, Arthur D Little found that in the IT industry the most innovative companies achieved up to twice the sales and profits, while also reducing break even time by half, compared with average companies. [See http://www.adl.com/uploads/tx_extthoughtleadership/ADL_InnoEx_Report_2010.pdf]

Similarly, a Report from October 2009 by the UK's National Endowment for Science, Technology and the Arts (NESTA) reported that "a small number of high-growth businesses are responsible for the lion's share of job creation and prosperity, and that innovation is instrumental in the success of these businesses…[and that]… Innovative firms grow twice as fast, both in employment and sales, as firms that fail to innovate…and the research shows that their innovation was a source of growth."

The Vital 6% - NESTA Report, 2009

[http://www.nesta.org.uk/library/documents/Vital-six-per-cent-Nov2010-v3.pdf]

End-user Innovation

An alternative to manufacturer-based innovation that has been increasing in prominence is known as end-user innovation. This is where an individual or group develops an innovation because they have a need for it (or they can see a community need for it). The obvious example of this (and becoming widely-used by software testers with tools such as Watir and Selenium) is open-source software. Many of us can now happily perform practically all our software tasks solely using open source software.

Innovation Relies on Innovators

Despite manufacturer-based and end-user based innovation each having quite distinct goals, both are absolutely dependent on the actual innovators. The results of an OECD 2008 Business Questionnaire showed that 93% of IT businesses agreed that the leading factor for encouraging innovation was the availability of suitably-qualified staff. Whichever form they are involved with it is likely that these staff are gaining by involvement in a challenging and absorbing job in which they are encouraged to learn and showcase new skills, thus making the role an attractive one.

> ***"Genius is 1% inspiration, 99% percent perspiration."***
>
> **Thomas Edison**

Genius Innovators?

So, do we need to look for geniuses to be our innovators? This would be a sad state of affairs for most of us who aspire to innovate. Luckily it appears that having a brain the size of a planet is not a pre-requisite. The consensus is that although genius would be useful, in practice it is hard work that plays the biggest part in succeeding as an innovator.

Multiple Innovations

Other than simply taking actual innovators words for it, there is further evidence that supports this 'hard work hypothesis' in the form of what are known as 'multiples'. This phenomenon has been understood for many years and it is where new discoveries are made practically simultaneously, but in isolation, by more than one scientist at one time. It seems unlikely that, given the scarcity of geniuses, two would be working on the same things at the same time by chance. Back in 1922 researchers documented about 150 such occurrences that were publicly known, one of the best known of which was the theory of evolution that was independently advanced by both Alfred Wallace and Charles Darwin at about the same time. It is also not difficult to imagine that there must have been many more 'multiples' that were never made public (many scientists have related in their private papers their annoyance at 'discovering' something only to later find that another had either got there first, or published first). The rationale for 'multiples' is that (in general) to make a new discovery not only does much hard work need to have been done, but the circumstances also need to be just right. Those circumstances normally include the requirement for a number of other supporting innovations to be in place, as well as the society the innovator is working in being conducive to the innovation (which many might supplement with "…and you have to be lucky, too.").

> ***"Too often we forget that genius, too, depends upon the data within its reach, that even Archimedes could not have devised Edison's inventions."***
>
> **Ernest Dimnet**

Adjacent Possibilities

The concept that certain innovations already need to be in place for a new innovation to be possible is sometimes known as 'adjacent possibilities'. Isaac Newton suggested he benefited from this concept when he talked about 'standing on the shoulders of giants'. An example of adjacent possibilities is provided by Gutenberg's invention of the printing press in 1440. A number of technologies had to be in place before the printing press could be invented. For instance, the ability to make suitable paper, the technology to develop an oil-based ink, and the concept of movable type (already used in China and Korea, but with clay type) were all necessary 'ingredients'. Gutenberg took these and applied his craft as a goldsmith to the already available concept of movable type to create movable metal type. The final piece of the jigsaw was that he lived in a part of the world (Germany) that was enjoying a boom in wine-making, thus allowing him to see the possibility of adapting wine press design for his new printing press. Who would not consider all these circumstances coming together for Gutenberg as lucky? Because, remember, the concept of 'multiples' suggests that if Gutenberg had not come up with it, then there's a good chance someone else would have done so fairly soon anyway.

> ***"To fully understand today's software testing we need to also understand the concepts on which it is built."***

Software Testing 'Lines of Innovation'

So, how does this apply to software testing? One approach is to look at the history of innovations in software testing and trace 'lines of innovation' that show (one interpretation of) how we have arrived at today's state of the industry. These models can help us to identify possible dead-ends, and, hopefully, indicate useful future trend lines. Probably more importantly, these lines of innovation show that today's approaches to software testing are not all new and cannot be understood in isolation of the innovations that came before them; to fully understand today's software testing we need to also understand the concepts on which it is built.

Core Testing Innovations of the 20th Century

Figure 1 shows what could be considered as the core software testing innovations of the 20th Century (a very inclusive view of innovations has been used here and you can see that I have started with innovations from even earlier). Innovations are typically made possible by a number of other previous innovations, but here only the most predominant are shown. The concept is very simple. If an arrow points from one innovation to another (e.g. 'Weinberg's Law' to 'Independent Testing') then this indicates that the first 'innovation' at the least acted as a partial catalyst in enabling the second innovation to take place. In places more than one innovation has been identified as a precursor to a later innovation, such as can be seen here with 'Life Cycle Testing (V Model)', which has four innovations feeding into it. Each of the innovations in the figure is briefly described in the corresponding Appendix.

'Lines of Innovation' between Testing and Quality

Figure 2 shows some of the major 'Lines of Innovation' between Testing and Quality. This figure illustrates a view of how the innovations of 'test certification', 'testing standards', 'test methodologies' and 'test process improvement' have been achieved. A number of innovations are repeated from the previous figure as there are naturally a number of common innovations.

It is recognized that these two models cover only a small part of the software testing discipline, and that there are many more innovations that could be included in lines of innovation up to the present day, such as cloud testing (presumably with cloud computing as a precursor), open source testing tools and context-driven testing. Tentative models have been created for these, but space and time constraints mean that these will be published at a later date.

Feedback Positively Welcome

Obviously this model of innovations and their connections is highly-subjective, and is based on an analysis of papers and articles, interviews with some (older) testers, and personal experience. If any reader has extra innovations that they feel could be added to the model, or know where the current model could be improved, I would be very happy to receive their feedback.

A Possible Future Innovation – People Testing

No paper introducing the concept of lines of innovation in software testing would be complete without at least one look at a possible future innovation for the discipline. Figure 3 introduces a possible future innovation called ‘People Testing’. This prediction is based on a number of recent innovations (crowd testing, virtual worlds, collaborative games and micro-volunteering) and an analysis of future trends across the world. A number of assumptions have been used, some more probable than others. For instance, it is assumed that over the next 30 years, the world’s population will continue to rise, and that, despite diminishing natural resources, the majority of this population will be housed and fed, due to technological advances, such as in food production. It is expected that many of these people will not enjoy a particularly high standard of living, nor will there be anywhere near full employment. This scenario could lead to a large number of bored, under-utilized people who look to escape their mundane lifestyle. One escape open to them, aided by the widespread availability of extremely high bandwidth internet, will be virtual worlds, such as ‘Second Life’, which are already very popular in a number of Asian countries.

One recent innovation has been collaborative online computer games, which, perhaps surprisingly, enjoy an approximate 3:1 advantage in popularity over online combative games. When combined with virtual worlds it is easy to imagine that virtual teams of people may well engage in collaborative activities in virtual worlds. The new innovation needed to bring these teams into the testing world will be the ability to translate real world software problems to this virtual world. If this can be done in a way that hides the technicalities (e.g. integrating enough monitoring to allow results to be recorded) and also ideally instils a sense of puzzle-solving to the situation then hopefully virtual teams of ‘people testers’ can be encouraged to address these ‘problems’ in the virtual worlds they visit.

Apart from the challenge of puzzle-solving, two approaches for encouraging people to take part are envisaged and the best solution may be to use a combination of both. First, there is a long history of individuals volunteering to take part in projects that are for the benefit of all. One of the latest innovations in this area has been micro-volunteering where mobile phone owners take part in experiments, where they give up just a short amount of time to a larger project. The other relevant innovation could be crowd testing, where testing jobs are crowdsourced to allow anyone who wants to attempt to test something to try and, typically, to be rewarded based on defects found. Thus the ‘people testers’ could be encouraged to address the testing puzzles in the virtual worlds by a combination of their innate curiosity, social conscience and rewards that could be used to improve their virtual experiences.

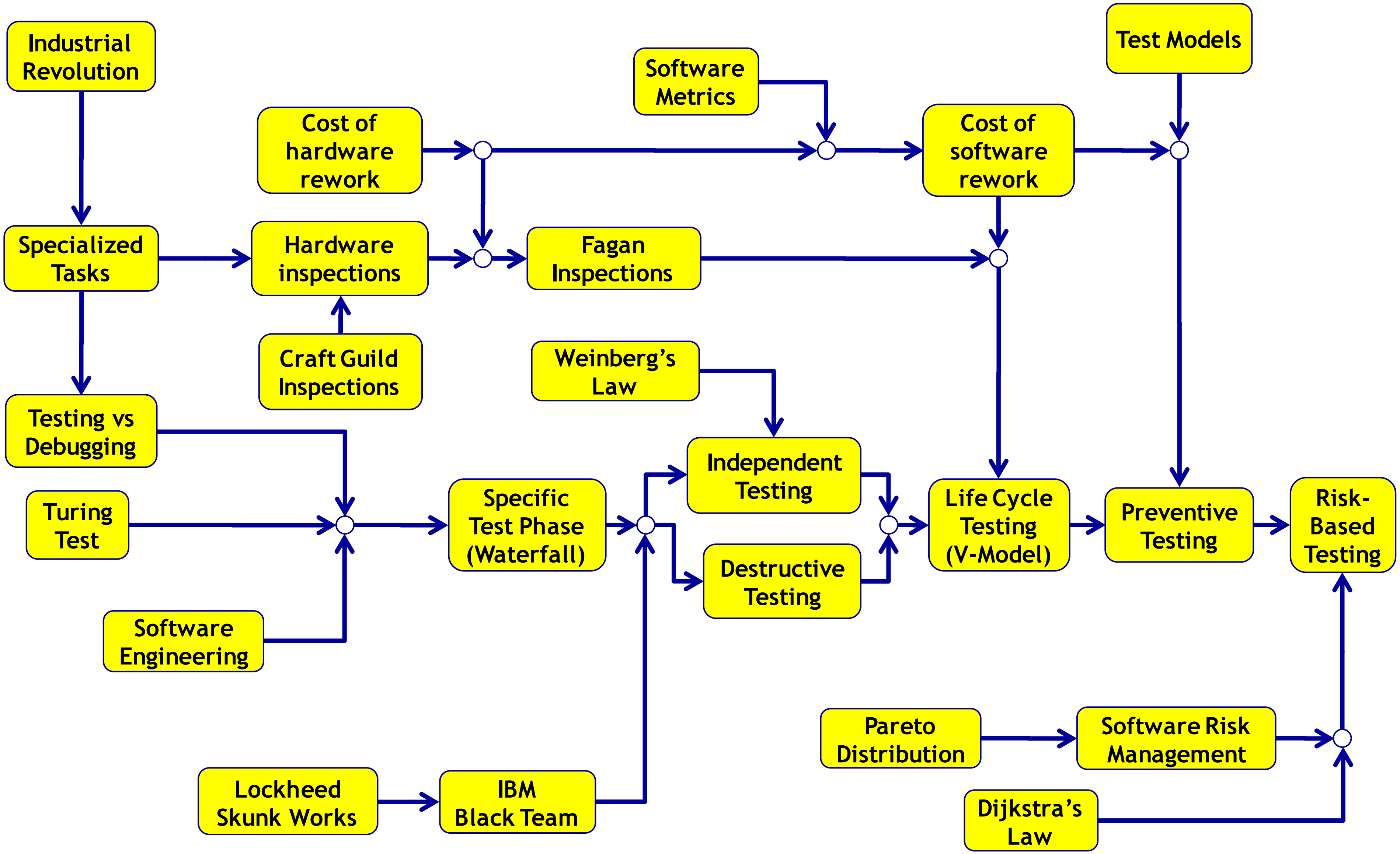

Figure 1: 'Core 20th Century Testing' Lines of Innovation

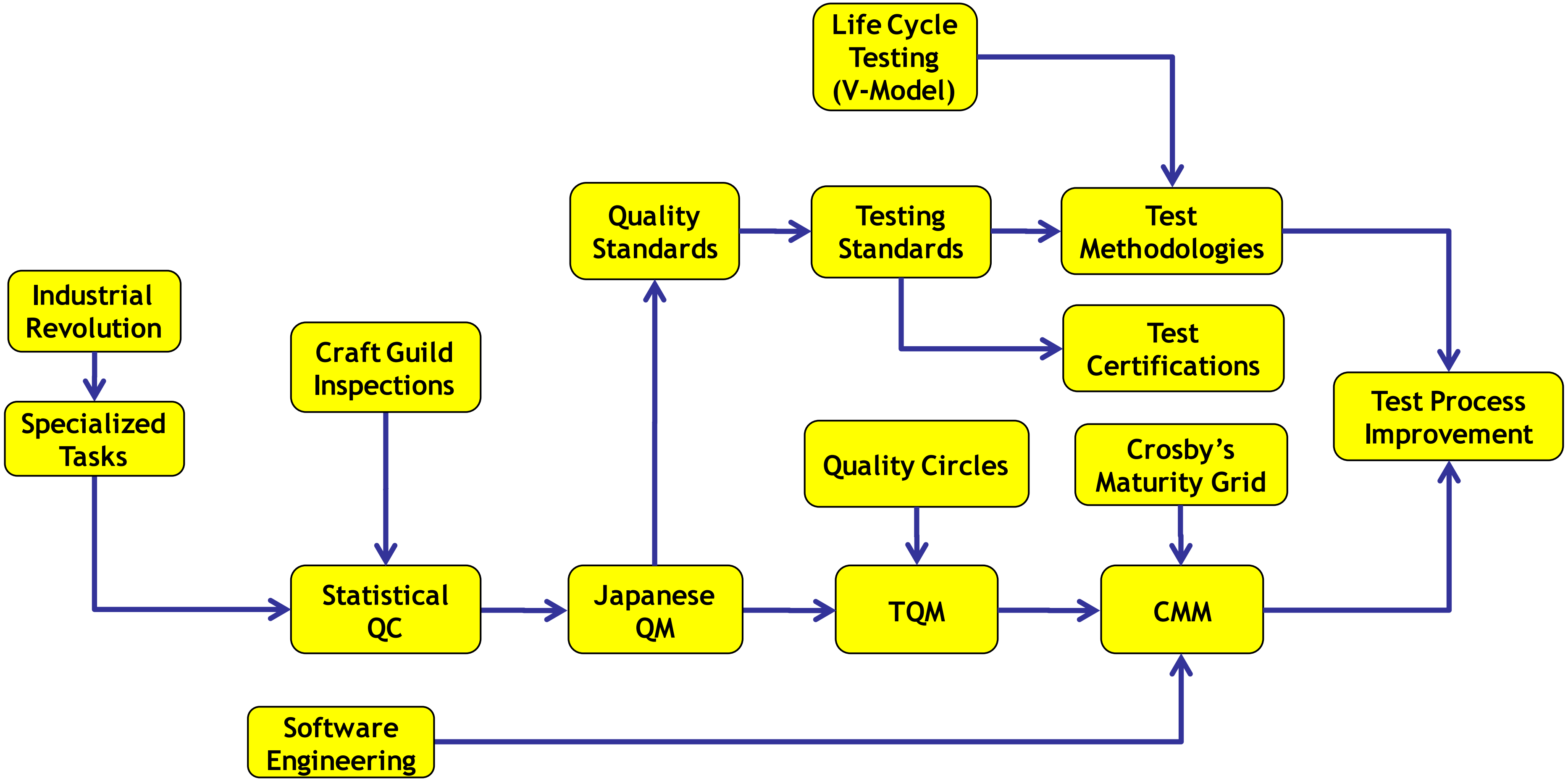

Figure 2: 'Testing and Quality' Lines of Innovation

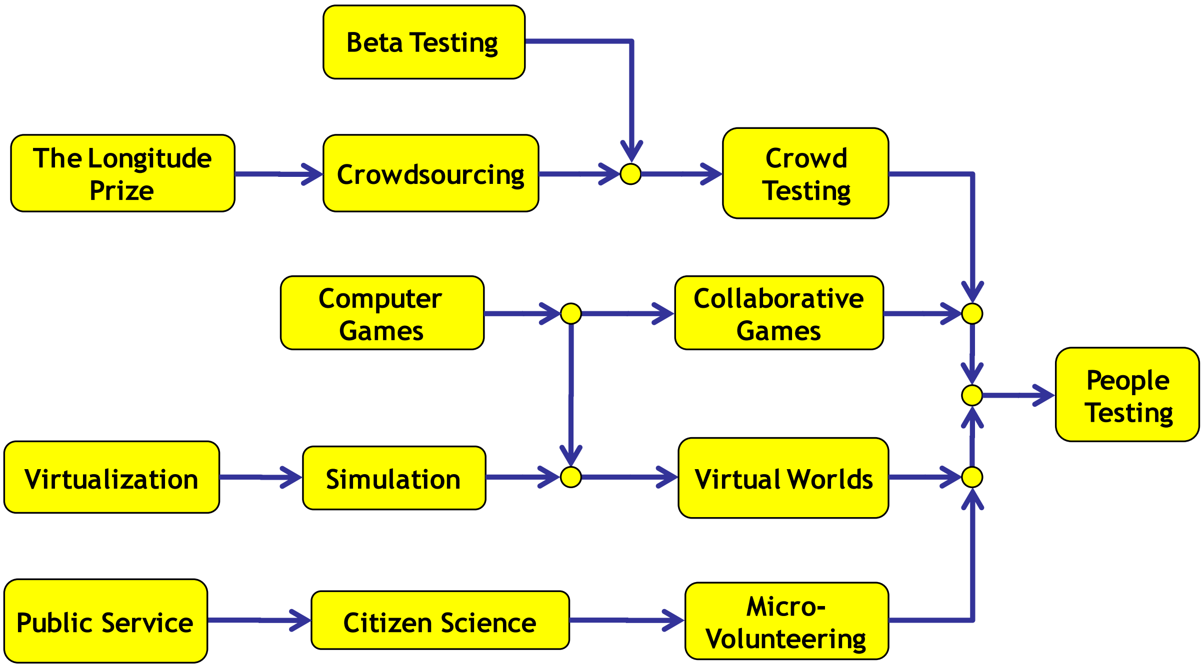

Figure 3: 'People Testing' Lines of Innovation

Encouraging Innovation

One benefit of analysing previous innovations is that it provides clues on the best approach for encouraging innovation. This information is useful both to organizations and individuals who want to innovate – and who wouldn't?

> ***"The Prime Factors for Innovation are: diversity, freedom, creative thinking, motivation, knowledge, and the physical work environment."***

There are a number of factors that affect an individual's (and so an organization's) ability to innovate. From the organization's perspective, addressing these factors should increase innovation and so thereby increase profitability, sales, etc. From an individual's perspective these factors should act as pointers to improving both their careers, and, more importantly, their job satisfaction. The main factors are: diversity, freedom, creative thinking, motivation, knowledge, and the physical work environment. We shall concentrate on the first three of these factors as they are specifically related to innovation. The remaining factors are fundamental to all testers' jobs and are covered extensively elsewhere.

Size Does Matter

Innovators do not work well in isolation. This is not news. If you study the history of innovation, you find that there appears to be a minimum population below which innovation is really slow. As cities emerged then innovations suddenly blossomed in them. Apparently when towns reach a 'critical' size then there are enough like-minded individuals to form groups that meet, discuss and bounce ideas off each other and, if the environment and focus are right (it is not just a matter of size), this can lead to a disproportionate increase in innovation.

Organizations can use this knowledge to encourage innovation in a number of ways. They can create physical working environments that promote discussion, and they can put enough innovators together to reach a critical mass. Where this is not possible, perhaps due to the small size of the organization, then they can provide their innovators with opportunities to talk to others, such as through conference attendance and encouraging their potential innovators to join user groups on the web, so that productive (and safe) discussion can take place.

> ***"That's simple … from differences. Creativity comes from unlikely juxtapositions. The best way to maximize differences is to mix ages, cultures, and disciplines."***
>
> **Nicholas Negroponte**

Diverse Team Members

The ability to find and talk with like-minded people raises a related diversity issue – innovators seem to work best in groups that are made up of dissimilar individuals. Thus we have to be careful not to create homogeneous groups of innovator 'clones'. It has been known for years that managers should resist the temptation to employ staff in their own image, and if we want our innovation groups to be more productive we need to create diverse teams with a range of ages, both sexes, different cultures and social groups represented. This team structure encourages innovation because this diversity provides team members with a variety of views and opinions. We can also expose our testing innovators to a wider variety of views by making sure we do not limit them to testing conferences and testing text books, but additionally encourage them to look outside the discipline.

> ***"If you have two people who think the same, fire one of them. What do you need duplication for?"***
>
> **Jerry Krause**
>
> As individuals we are prone to seeking out those who agree with our point of view and although innovators need the safety of discussion in friendly groups, it is also important that we take care to avoid groupthink. This is where the group lose the ability to critically analyse the ideas they are uncovering and can also lead to the group's position becoming more radical as individual members vie to demonstrate their group 'credentials'. Innovation thrives best when ideas are thoroughly examined by sceptics.

> ***"Expose yourself to the best things humans have done and then try to bring those things into what you are doing."***
>
> **Steve Jobs**

Look Outside of Testing

The approach of looking outside the testing discipline for new ideas that can be applied to testing has a long pedigree. Over the years many innovations have come about by successfully introducing concepts from one field to a completely different field, and we have already seen an example of this with the modification of the wine press into a printing press by Gutenberg. As Thomas Edison points out to would-be innovators: "Keep on the lookout for novel ideas that others have used successfully. Your idea has to be original only in its adaptation to the problem you're working on." The application of concepts from Far Eastern martial arts to software testing (e.g. Testing Dojos – see http://www.testingdojo.org/tiki-index.php?page=Testing%20Dojos) is an example of such an approach.

Freedom

> ***"The best way to have a good idea is to have lots of ideas."***
>
> **Linus Pauling**

The freedom to make mistakes is essential to successful innovation. We typically need to try and fail and try again many, many times before we succeed. Tom Peters, management guru, and author of 16 books, explains *"Do it right the first time is insane advice. Nobody does anything…INTERESTING…right the first…or the twenty-first…or the forty-first time. Doing the new means screwing around, trying stuff, and messing stuff up again and again and again."*

> ***"I have not failed, not once, I've discovered a thousand ways that don't work."***
>
> **Thomas Edison**

Embracing Failure

Innovation is based on embracing testing and failure. We know that we are not going to come up with the 'next big thing' on our first attempt, and that means that, *realistically*, we must innovate within a rigorous testing regime that expects failure and expects repeated attempts to improve. Actually it appears that working in perfectionist regimes like Six Sigma and TQM (Total Quality Management) may hamper innovation as the overall quality paradigm and the lack of acceptance of defects may act as a constraint on innovators. Does this mean that quality and innovation are mutually exclusive? Obviously not (we are not going to buy low quality innovations), and as testers we are probably best placed to see how finding defects contributes to the development of higher quality products. To be an innovator means you have to accept that failure is part of your daily life, and you must also have the determination and belief to go on until you achieve the success of creating a true innovation. Organizations need to accept that innovators need that level of freedom. In some respects innovators are like babies - they need to be allowed to explore and try things and continually fail, but to eventually succeed they need to be allowed to do this in a nurturing environment.

> ***"We offer our engineers '20% time' so that they're free to work on what they're really passionate about"***
>
> **Google Jobs Page**

Innovation Time!

Some organizations, Google and 3M being perhaps the most well known, implement a strategy that explicitly provides a degree of freedom for their staff to innovate. Google (20%) and 3M (15%) allow their engineers to spend a proportion of their time on projects that the engineers really want to do. At first glance this appears to be rather indulgent and perhaps only possible for organizations that have spare capacity and lots of money in the bank to start with. A closer look at the approach, however, contradicts the idea that this is an extravagance when the benefits are considered. First, such a strategy contributes greatly in persuading staff that their ideas are valued and so increases the flow of potential innovators into the organization and reduces the flow of innovators away from the organization. Second, and maybe more importantly to the accountants, this strategy is really successful in generating innovations. We nearly all use Post-It Notes, a result of the 3M 15% rule, and Google estimate that approximately 50% of their new products come out of the 20% innovation time. One example, AdSense from Google, currently generates approximately 30% of their revenue (it allows website owners to earn revenue by displaying (relevant) Google advertisements).

Creative Thinking

Creativity is a core skill for innovators, however the best way of encouraging people to think creatively is very subjective and correspondingly difficult to measure. Many of us feel that we do some of our most creative thinking in the shower, or perhaps walking beside a river – in fact, anywhere other than sitting at a desk. This approach allows our brains to associate disparate ideas by taking us away from the focused work of our daily lives. Many innovators describe how their inspiration came from dreams, which scientists have now confirmed as a reasonable explanation. Apparently while in REM sleep many new circuits in our brains 'fire' and occasionally these new circuits will bring together previously distinct ideas that form the beginnings of a solution that on waking, if remembered, can become the seed of an innovation. However, no-one appears to be advocating, as yet, that we should encourage our innovators to spend half their day asleep in the hope that their dreams will be fruitful.

Get a Notepad

Personally, although the shower and riverside walks work well for me, I find that as I get older the association of ideas needs some help, which I find is best provided by using a low-tech notepad. By carrying this with me I can record my (and other peoples') ideas and by occasionally reviewing them I can (sometimes) come up with useful ideas myself.

> ***"In the field of observation, chance favours only the prepared mind."***
> ***- Louis Pasteur***

Serendipity

This lucky association of ideas when we are awake, day-dreaming or in REM sleep is a form of serendipity – where we come across a possible new solution by a lucky combination of accident and wisdom (wisdom, because we have to recognize how the thing we have come across by accident is useful – we have to be 'prepared'). We have already discussed increasing our chance of 'lucky' accidents by increasing the diversity of our working environment, by reading outside of the testing area and by going to non-testing conferences, but that is only a partial solution. We need to also address the achievement of wisdom. We can improve this from an organization's perspective by ensuring we employ suitable individuals, while both individuals and organizations should take responsibility for their continued professional development.

Conclusion

This paper has introduced a number of the concepts behind innovation and has provided evidence on why today's businesses should be embracing a culture of innovation even while in a recession. The concept of 'adjacent possibilities' has been extended to create 'lines of innovation' and this has allowed a number of the innovations that form the basis for software testing to be modelled in these terms. An attempt has been made to 'stretch' a 'line of innovation' into the future to provide a prediction of a future testing innovation – 'people testing'. Finally, the prime factors for promoting innovation (diversity, freedom, creative thinking, motivation, knowledge, and the physical work environment) were identified and the means of achieving the first three were considered in some detail.

The ‘lines of innovation’ models for software testing provided in this paper are necessarily a (subjective) first attempt, but feedback and additions to the models are requested.

Appendix A – Core Testing 'Lines of Innovation' 20th Century

This appendix provides brief descriptions of the innovations included in Figure 1 (in alphabetical order).

Cost of Hardware Rework

As described by Michael Fagan, the result of discovering hardware errors in silicon chips was to scrap the faulty pieces, re-design and then re-manufacturer resulting in enormous rework costs. It is perceived by many to be easier to fix software, but the costs can still be very high depending on when the faults are detected.

Joseph Juran and Philip Crosby believed that by preventing defects, the cost of rework would decrease by far more than the prevention cost; this argument formed that basis for Crosby's 1979 book, 'Quality is Free'.

Cost of Software Rework

A number of researchers in the early 1980s published books (e.g. Barry Boehm's 1983 'Software Engineering Economics') and papers (e.g. US National Bureaus of Standards' 1983 Guideline for Lifecycle Validation, Verification and Testing of Computer Software) outlining the costs of detecting faults late in the life cycle. Many books, syllabi and papers still use the figures reported at this time as the justification for testing as early in the life cycle as possible.

Craft Guild Inspections

An early form of quality control from the middle ages, introduced to maintain defined standards, protect consumer rights, and as a form of protectionism for the guild members from the import of cheaper 'foreign' goods.

See 'The later middle ages: Craft organisation and the Guilds', A History of the County of York: the City of York (1961), pp. 91-97. URL: http://www.british-history.ac.uk/report.aspx?compid=36334 Date accessed: 15 July 2014.

Destructive Testing

Destructive testing was advocated by Glenford Myers in his 1979 classic book, 'The Art of Software Testing' and this led to a generation of testers aiming to 'break' the software. Myers believed that by attempting to demonstrate that something was working, then 'easier' tests would be chosen in order to succeed in this goal. His response was to suggest that testers' main purpose should be to 'break' the software they were testing, very much aligned with the approach followed by the IBM Black Team.

Dijkstra's Law

Testing can show the presence, but not the absence of defects.

In 1970 Dijkstra expounded his rather negative, but realistic 'law'. He argued that due to the near infinite number of possible tests for any real software then running all these tests (exhaustive testing) is impossible; hence it is always possible that a defect corresponding to one of the unexecuted tests can remain undetected. Thus, as testers, we are faced with the task of choosing the best possible subset of this exhaustive set.

Fagan Inspections

Based on hardware inspections, Michael Fagan published his seminal paper, 'Design and code inspections to reduce errors in program development' in the IBM Systems Journal, in 1976. Fagan Inspections have the dual goals of finding faults both in the product and in the process used to create the product. They can be applied to any artefact and have been measured to be probably the most effective technique in improving software quality.

Hardware Inspections

Fagan describes the use of design inspections of silicon chips to supplement the dynamic testing of the chips prior to release of the design into production. This was in the 1960s in IBM.

IBM Black Team

The IBM Black Team was created in the 1960s by assembling those engineers who seemed better than their peers at finding defects into a team that was only concerned with testing. When in a team their efficiency increased dramatically and they bonded into the 'black team', who tried to break the developers' code (hence the destructive testing). They were known as the black team as they wore black t-shirts as an informal uniform.

Independent Testing

Independent testing is testing performed by someone distinct from the developer of the software under test. Levels of independence for software testing were included in BS 7925-2:1998.

Life Cycle Testing

Dating from 1979, the 'V' life cycle model explicitly shows the relationship between development and testing activities, but perhaps more importantly it also explicitly identifies the output of each phase and suggests that this should be reviewed to ensure it is good enough to be used as the input to the next phase. Thus static testing is introduced into this life cycle model as soon as anything is developed, and testing is now an activity across the whole life cycle.

Lockheed Skunk Works

Set up in 1943 to work (sometimes without a contract) on rush orders for the war effort, resulting in some spectacularly successful aircraft, such as the P38 Lightning in WW2, the U2 spy plane and the F-117 stealth fighter of the 1980s, made famous in the first Gulf War.

The Skunk Works (see http://www.lockheedmartin.co.uk/us/aeronautics/skunkworks.html) used fourteen working principles, among which they advocated a lean inspection system that required suppliers to perform inspections and test early. The use of small, protected teams was an integral part of the approach.

Pareto Distribution

Just over 100 years ago the Italian Vilfredo Pareto identified his eponymous 80%-20% distribution. When applied to testing it has been interpreted as: (*Approximately) 80% of defects are found in (approximately) 20% of the code*. Although not always 80-20, there is plenty of evidence that backs up the assertion that most of the defects are found in a small proportion of the code.

Preventive Testing

In 1988 David Gelperin and Bill Hetzel propose a new 'prevention testing methodology' in which they describe a 'test sharing strategy' in which programmers are provided with the tests before they write the code, thereby reducing defects and rework.

Risk-Based Testing

Even though risk-based testing is a natural extension of software risk management, it took until the late 1990s for risk-based testing to get to the level where using risk to derive test strategies was regularly practiced. The more 'advanced' concept of using test results to inform ongoing risk analysis is understood, but not currently widely-practiced. Risk-based testing is now the basis for the new ISO/IEC 29119 Software Testing Standard (parts 1-3 are published and 4-6 are in progress).

Software Engineering

The emergence of the requirement for software development to be performed as an engineering discipline (at a NATO software conference in 1968), was partly based on the identification by the US Department of Defense of a 'software crisis' (see http://en.wikipedia.org/wiki/Software_crisis), based on their experiences with defence contractors.

Software Metrics

Basic software metrics (i.e. lines of code), have been collected and/or used since the late 1960s as measures of software quality and complexity. In the mid-1970s more measures of complexity, such as McCabe's Cyclomatic Complexity and Halstead's Measures became available, along with Function Points.

(See http://www.eecs.qmul.ac.uk/~norman/papers/new_directions_metrics/HelpFileHistory_of_software_metrics_as_a.htm)

Software Risk Management

The field of risk management has been around in the financial and insurance worlds for centuries and started to be (explicitly) applied to software in the mid-1980s. Two leaders stand out in this field. Bob Charette published his book 'Software Engineering Environments: Concepts and Technology' in 1986 and was subsequently employed full-time as a risk guru. Barry Boehm also published his Spiral life cycle model in 1986, which was based on managing requirements risk through prototyping, and later in the 1980s published a set of risk tutorials for the IEEE.

Specialized Tasks

In the early 20th century, F W Taylor introduced the idea of 'scientific management', which included the idea of time and motion studies. He believed that making a job more efficient required breaking it down into its component parts and addressing each part separately. This concept allows the separation of testing (initially quality control) from the actual product creation (see http://en.wikipedia.org/wiki/Frederick_Winslow_Taylor).

Specific Test Phase

In 1970 Winston Royce first describes a life cycle model (now known as Waterfall) in which 'testing and debugging' is identified as a separate life cycle phase (from requirements, design, coding, etc.). Note that he presented the model as being flawed and not useful unless used in an iterative manner.

Test Models

As early as 1984 academics were proposing the idea that abstract test cases (test models) applied to requirements and design specifications can find defects in these specifications very early in the life cycle, thus preventing defects in these specifications and so reducing rework costs.

Testing versus Debugging

Apparently a 1957 review of the book Digital Computer Programming by Dan McCracken distinguished the activities of testing (finding the fault) from debugging (fixing it). There are, however, earlier examples of this division being employed. For instance, in 1954 the programmers and engineers (testers) working on the world's first business computer, LEO, for J. Lyons and Co in the UK, were told by the project manager, who apparently had strong views on the division of labour, that engineers would detect faults, while the programmers would fix them.

See 'User-Driven Innovation: The World's First Business Computer', by Caminer, Aris, Hermon and Land, McGraw-Hill, 1996.

Turing Test

Alan Turing wrote a paper on Artificial Intelligence (AI) and proposed an 'imitation game' (now known as the 'Turing Test') to be used to decide if AI had been achieved. In the game/test a tester asks questions (tests) of both a human and a computer and decides from their answers which is which. If the tester cannot tell, then the computer can be assumed to be 'intelligent' (as we typically judge other people's intelligence from their responses to questions).

Alan Turing's 1950 paper is 'Computing Machinery and Intelligence' in Computing Machinery and Intelligence (Mind 59). See http://www.turing.org.uk/turing/scrapbook/test.html.

Weinberg's Law

A developer is unsuited to test his or her code.

In 1971 Weinberg argued that a developer is often blinded to their own mistakes because if the assumptions they made while developing were wrong then by making the same assumptions while testing they will not discover the corresponding defect. This type of defect can normally only be found by someone else.

Appendix B – Testing and Quality 'Lines of Innovation'

This appendix provides brief descriptions of the innovations included in Figure 2 that were not already described in the previous appendix (in alphabetical order).

CMM

Development of Watts Humphrey's Capability Maturity Model (CMM) began in 1986 at the Software Engineering Institute at Carnegie Mellon University funded by the USAF. It was intended as a means of evaluating the suitability of contractors developing software for the US Department of Defense, but has since been expanded to cover a number of other areas. It comprises five levels of maturity (Initial, Repeatable, Defined, Managed and Optimizing). CMM evolved into CMMI (Capability Maturity Model Integration), which was published in 2002.

See Watts S. Humphrey's 'A Method for Assessing the Software Engineering Capability of Contractors', CMU/SEI-87-TR-2, 1987.

Crosby's Maturity Grid

One of the earliest organizational maturity matrices is Philip Crosby's Quality Management Maturity Grid (QMMG), which describes the typical behaviour exhibited by a firm at five levels of 'maturity'. This grid suggests that companies are likely to evolve through five phases - Uncertainty, Awakening, Enlightenment, Wisdom, and Certainty. The Quality Management Maturity Grid was first published in Crosby's book 'Quality is Free' in 1979 (Philip B. Crosby, Quality is Free, New York, McGraw-Hill, 1979).

Japanese QM

After the end of the Second World War, as part of the reconstruction support to Japanese industry, Dr. W. Edwards Deming visited many times (along with others) and he was invited to teach on statistical quality control (Walter Shewhart, their first choice, was ill). His contributions to the improvement of quality management made a lasting impression on Japanese industry.

Quality Circles

Quality Circles were initially used in the early 1960s in Japan, and were derived from the work of Dr. W. Edwards Deming and Joseph Juran on quality management. They advocated that the workers (rather than the management) should be given more responsibility for quality, thereby increasing employee participation in decision-making and quality improvement. Quality Circles comprise small groups of workers who consider the work they do and try to improve the quality of the product they are building and the process they use to build it.

Quality Standards

The US Department of Defense identified the requirement for a quality standard after their experiences with seriously deficient material provided for their forces during the Second World War and the Korean War. This resulted in Mil-Q-5823 and Mil-Q-9858 but these were found to be too prescriptive and were quickly superseded by Mil-Q-9858A, published in 1959, which was far more based on end results (see http://www.datanalysis.co.uk/industrial_career/rr_career/manufacturing/9_chief_quality_engineer/milq9858a_and_sons.htm).

Statistical QC

Statistics were initially practically applied to Quality Control in the 1920s, notably by Walter Shewhart of Bell Telephones, who was looking at the quality of the microphones in telephone handsets. He later published a book on the topic (*Economic control of quality of manufactured products,* New York, Van Nostrand, ISBN 0-87389-076-0, 1931) and became known as the 'Father of Quality Control' (see Denis Bayart, *The fact–theory dialogue in an industrial context: the case of statistical quality control*, European Management Review (2006) 3, 87–99. doi:10.1057/palgrave.emr.1500055).

Test Certifications

QAI started the CSTE scheme in the US in the mid 1990s – this required no examination at the time, but has since adopted an exam-based approach.

The ISEB testing certification scheme started in 1998, and the working group was comprised of many of those who had worked on the British Standards on Component Testing and Vocabulary of Testing Terms (BS 7925). The ISTQB scheme was derived from the ISEB scheme and started in 2002 – the two schemes have since 're-aligned' as far as testing is concerned and there were over 336,000 testers who have successfully achieved some level of certification (as of late 2013).

Testing Standards

Work on the first testing standard, IEEE 829 Software Test Documentation, began in 1979 and it was published 4 years later. Subsequently the IEEE published a Unit Test Standard (IEEE 1008) in 1987. In 2007 ISO/IEC started work on their first Software Testing standard, ISO/IEC 29119 (parts 1-3 are published and 4-6 are in progress).

Test Methodologies

David Gelperin and Bill Hetzel created the STEP ('Systematic Test and Evaluation Process') software testing methodology, basing it on the IEEE Unit Testing Standard, which was published in 1987. In Europe Sogeti published its test methodology, TMap, in 1995.

Test Process Improvement

Test process improvement approaches are generally dependent on a combination of a model of the test process and an assessment method (e.g. ISO 15504 SPICE). Probably the first test process improvement 'approach' was the Testing Maturity Model (see Burnstein, I., et al, Developing a Testing Maturity Model: Part 1 and 2, Illinois Inst. of Tech., 1996), which was based on CMM and created due to CMM not covering testing to any great extent. Subsequently a number of approaches/models have been published, some commercial and some not-for-profit, such as TPI®, TMMi, etc.

TQM

The term 'total quality' was first coined in 1969 and subsequently the core ideas of total quality management (TQM) set forth by W. Edwards Deming, Joseph Juran, and Kaoru Ishikawa gained acceptance in the U.S. management community. The focus of TQM widened from quality of products to quality of all issues within an organisation, including the involvement of all employees, customer focus, continuous improvement and the integration of quality management across the whole organization.

Appendix C – 'People Testing'

This appendix provides brief descriptions of the innovations included in Figure 3 (in alphabetical order).

Beta Testing

A form of acceptance testing for 'shrink-wrapped' software, where the pre-release system is provided to 'tame' customers (or sometimes even the general public) for them to test the system in their environment. This extends the amount of acceptance testing and, perhaps more importantly, increases the range of test environments.

Beta testing was initially a hardware testing term and has been adapted to reflect a slightly different approach in software (see http://en.wikipedia.org/wiki/Software_release_life_cycle).

Citizen Science

Citizen science is where volunteers, who may be untrained, perform scientific research on a topic of interest to them. Examples include bird surveys, one of which started in 1900, the community creation of the Oxford Dictionary in the 1920s, and also SETI (the search for extraterrestrial intelligence), although in this latter example the volunteers' computers do most of the work. Citizen science has changed and expanded as the internet makes communication between the citizen scientists easier.

Collaborative Games

Collaborative online games are now an extension of PC games, but where the players interact with each other across the internet. Initially these were known as MUD (multi-user dungeon) games and the first came out in about 1975 running on a DEC PDP-10, and by 1977 there were MUD games running across the ARPANET. Nowadays, where many players use the same environment then these games are often known as massive multi-player online (MMO) games. The most popular MMO in mid-2013 is 'World of Warcraft', with an estimated 7.7 million players worldwide. It is said that collaborative games are now three times as popular as competitive online games.

Computer Games

Computer games have been around for practically as long as there have been computers, but in the early days the games were built by hobbyists mainly at universities. In the mid-1960s a number of patents were awarded for video gaming in the US. Commercial arcade games became available in 1971 with Atari's Pong being launched in 1972. Home computer consoles (and games) came out in the late 1970s, and subsequently there have been a variety of other platforms, including PCs, handheld computers (e.g. Nintendo DS) and mobile phones. In 2013, Grand Theft Auto 5 took about £500 million on its first day, easily recouping the estimated £170 million development and marketing costs.

Crowdsourcing

This term was coined in June 2006 by Jeff Howe in Wired Magazine in his article 'The Rise of Crowdsourcing' (see http://www.wired.com/wired/archive/14.06/crowds.html); the idea being for companies to ask the general public (the crowd) to solve their problems for them when they do not have the resources or skills to do it themselves. An example provided by Howe is that of InnoCentive, who use a group of some 90,000 'solvers' to provide solutions (rewarded from $10,000 to $100,000) for a range of well-known organizations (known as seekers) at a success rate of more than 30%.

Some commentators take an even wider view and consider implicit demands to be crowdsourcing, and in that case YouTube and Wikipedia could be considered as providing crowdsourced video content and information, respectively.

Crowd Testing

A natural extension to crowdsourcing, crowd testing is a specialized form of crowdsourcing where the testing of a system/application is opened up to the general public. Typically this form of testing is web-based, although there is also a specialist branch concerned with testing mobile devices. Crowd Testing uses two basic approaches to rewarding their 'crowd'; one is a per defect approach, while the other is an hourly rate.

The Longitude Prize

In the early 18^{th} century the British Government were looking for a solution to the problem of accurate timekeeping at sea as this would allow ships to determine their longitude, and so lead to fewer losses. They had found that the traditional approach of contracting a solution was not working and so launched a public prize. In response a number of inventors decided to solve the problem, and in 1714 John Harrison won the considerable prize of £20,000 (equivalent to over €4M today) with his marine chronometer. The prize was awarded even though Harrison was 'only' the son of a carpenter and was an unexpected winner.

Micro-Volunteering

Micro-volunteering is where an individual chooses to spend up to 30 minutes for a good cause; they may be particularly skilled in the area they work in, or may simply be using their 'common sense'. This time may be used to complete a task, or it may be just one part of a larger task being worked on by many micro-volunteers. An example of a micro-volunteering task may be where a not-for-profit organization needs a new marketing leaflet reviewed. Most micro-volunteering is performed either through personal computers or mobile phones across the internet. Micro-volunteering by phone started in 2008 with 'The Extraordinaries' and their initiative is now managed through their Sparked.com website.

Public Service

Volunteer public service (aka community service) is where volunteers donate their time to perform activities for the public good. Public service is typically voluntary and not directly compensated, but in some cases it may be rewarded in some way, although typically below its market value.

Public service can be considered as a form of philanthropy. Philanthropy originated in Ancient Greece and today is normally considered to be the wealthy donating their money, but can also apply to anyone who gives their time for the public good.

Simulation

Simulation is the imitation of a real thing (e.g. a process or system), and entails the representation of the most important characteristics of this thing.

The Monte Carlo Method of statistical sampling to model problems was initially discovered in 1777 by the Comte de Buffon with his 'needle experiment' on geometric probability and subsequently coined by the scientists working on the Manhattan Project in 1946, who used early computers to simulate neutrons travelling through materials to help design radiation shielding.

Simulation is now used in many different areas for, among others, design, optimization, training, education and entertainment when the real object of the simulation, for instance, doesn't yet exist, is difficult to get access to, or is too dangerous to use.

Virtualization

Virtualization is the creation of a virtual (rather than actual) version of something (e.g. an imitation/simulation on a computer), such as a hardware platform, operating system, a storage device or network resources. It was initially used in the 1960s to increase the efficiency for multiple users of large mainframes. For instance, virtual memory is 'created' by operating systems when there is not enough physical RAM (random access memory) for all the running applications by using some of the memory on the hard disk and 'pretending' it is RAM. With network virtualization the users of the network make use of the available resources available on the network, but will typically not be aware of the actual physical network structure, nor of the actual number and type of devices attached to it.

There is a subtle difference between virtualization and simulation. Virtualization is more to do with making the most efficient use of resources (and hiding complexity from the user), while simulation is more about one system 'pretending' to be another system.

Virtual Worlds

A virtual world is a computer-based simulated environment which hosts an online community, who typically interact with the 'world' and the other users via avatars. As virtual worlds have evolved there are some that deliberately mimic the real world, which makes the cross-world movement of information and practices easier – these are sometimes known as mirror worlds.

According to Wikipedia the first virtual world was Habitat, developed in 1987. There are now several hundred virtual worlds accessible across the internet. Strategy Analytics report a 23% compounded annual growth rate for virtual worlds from 2009-2015, with in-game microtransactions expected to grow to £10 billion by 2015.

Second Life is a well-known virtual world for adults, which was launched in 2003 and has over 38 million users registered in 2014, although Habbo, which originated in Finland in 2000 had some 273 million (mostly teenage) avatars registered by 2012. Happy Farm, which is very popular in China, has approximately 230 million active users and 23 million daily users at the height of its popularity in 2010.