Multifractal Aspects of Software Development (NIER Track)

Abram Hindle
Department of Computer Science
University of California, Davis
Davis, California, USA
ahindle@softwareprocess.es

Michael W. Godfrey
David Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, CANADA
migod@uwaterloo.ca

Richard C. Holt
David Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, CANADA
holt@uwaterloo.ca

ABSTRACT

Software development is difficult to model, particularly the noisy, non-stationary signals of changes per time unit, extracted from version control systems (VCSs). Currently researchers are utilizing timeseries analysis tools such as ARIMA to model these signals extracted from a project's VCS. Unfortunately current approaches are not very amenable to the underlying power-law distributions of this kind of signal. We propose modeling changes per time unit using multifractal analysis. This analysis can be used when a signal exhibits multiscale self-similarity, as in the case of complex data drawn from power-law distributions. Specifically we utilize multifractal analysis to demonstrate that software development is multifractal, that is the signal is a fractal composed of multiple fractal dimensions along a range of Hurst exponents. Thus we show that software development has multi-scale self-similarity, that software development is multifractal. We also pose questions that we hope multifractal analysis can answer.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management—*Lifecycle*

General Terms

Measurement

Keywords

multifractal, fractal, version control, wavelets, power-law

1. INTRODUCTION

In this paper we argue and demonstrate that changes per time-unit signals, extracted from a software project's version control system, are often generated from complex stochastic processes that are not easily modeled via commonly used time-series analysis techniques. We demonstrate this fact via evidence that shows that changes per time-unit signals extracted from version control have multifractal properties such as fractional dimensionality and self-similarity.

ICSE '11, May 21–28, 2011, Waikiki, Honolulu, HI, USA

Various signals in software engineering have been studied using time-series analysis [5]. These signals include bugs per time unit, mailing-list messages per time unit and version control revisions per time unit. A common feature that these signals share is that they often are drawn from power-law [8], exponential, log-normal or Pareto [6] distributions.

Jingwei et al. [8] observed that many software engineering related signals like changes per day exhibit power-law-like tendencies. This power-law behaviour indicates that there are multiple scales to the signal, and that the signal is potentially self-similar at different scales. This means that zoomed in views of the signal will look similar to zoomed out views of the signal.

Because of this observation we know that other models popular in time-series analysis such as the Box-Jenkins ARIMA model [1, 5] are potentially inappropriate due to their assumptions about the data they analyze. Some of these assumptions are that the data presents homoscedasticity (constant variance), it is stationary (constant mean and variance), and has normally distributed errors.

This is relevant to software development because self-similarity is inherent to how we as a community have modeled modern software development. Software development is composed of repeating patterns of iterations. These iterations are broken in to phases, disciplines and activities. These different scales of process are interesting because they suggest multi-scale similarity might exist at those scales. Thus it seems that software development is seemingly composed of repetitions at different granularities. Therefore software development has its own intentional elements of self-similarity.

Why must we bother with this technique? We must address the complexity of software development activities if we want to model them. Modeling these activities allows for multiple uses such as improved predictions and estimations about change, and a better understanding about development processes that emerge from prescribed and ad-hoc processes. Furthermore we need models to handle the noise within these signals, as it is the noise which defines these signals.

One intriguing aspect of this analysis is if the dimensionality of a signal can give us indicators of the health or the stability of a system and its development. One of the compelling examples of multifractal analysis is the application of multifractal analysis to electrocardiograms. P.Ch. Ivanov et al. [7] found that the range of dimensionality of electrocardiograms was an indicator of the health of a human heart (see Figure 1). Can this kind of discovery be applied to soft-

ware development? What if the dimensionality of a project's development signals collapse near release time or due to an increasing system complexity?

We wish to investigate if various software development signals are multifractal. Although in this paper we investigate and demonstrate that a single software engineering relevant signal, changes per time unit, is often multifractal. We then conclude with various suggestions for future directions and future work that utilize this kind of analysis.

Our contributions in this paper include:

- We demonstrate that many software development behaviours exhibit multifractal properties.
- We provide a software engineering relevant introduction using multifractal analysis to the software engineering research community.
- We confirm that changes per time unit signals extracted from version control exhibits fractal properties beyond just being a power-law.

1.1 Previous Work

With respect to software engineering research Jing Wei Wu [8] described software development processes as fractal because many explicitly matched power-laws. Power-law-like signals are not necessarily multifractal but are definitely fractal and exhibit fractal scaling.

Marco D'Ambros created a "fractal visualization" that utilized scaling techniques to create zoomable scaling views [3].

1.2 Self-similarity and Fractal Behaviour

Multi-scale self-similarity is the similarity of a signal to itself at multiple scales. That is if you normalize for scale, data at one scale will look similar to data at another scale. An example from nature of this phenomena is that small streams and creeks often mimic the shapes that larger rivers have but at much grander scales. Fern leaves have fronds which scale from the very small to the very large while maintaining a similar overall shape. If you normalized either of these, rivers and streams or fern leaves, you would find that the entities are similar once normalized, thus these similarities exist at multiple scales. These similarities are more dramatically depicted in Mandelbrot fractals (see Figure 2) where the large shape of the fractal often appears at smaller scales.

Self-similarity is often considered to be inherent to fractal behaviour. Mandelbrot adapted fractals to timeseries analysis, eventually he discovered that some signals were composed of more than one fractal or a range of fractals, thus creating multifractals. To address this kind of signal, multifractal analysis was created. This is a kind of analysis that attempts to determine if a signal is multifractal and what kind of self-similarity occurs.

In Figure 1 we can see the plot of the dimensionality of the electrocardiogram of a healthy heartbeat (the large lump) and an unhealthy heart beat (the small lump). The $D(h)$ plot, explained in Section 1.3, indicates the range of fractal dimensionality of a signal [7] over its range of Hurst exponents. The smaller the range and domain of $D(h)$ the less likely that the signal exhibits multifractal behaviour. This example might have an analogue within software development. Perhaps when the dimensionality of development collapses an externally motivated or catastrophic event is occurring.

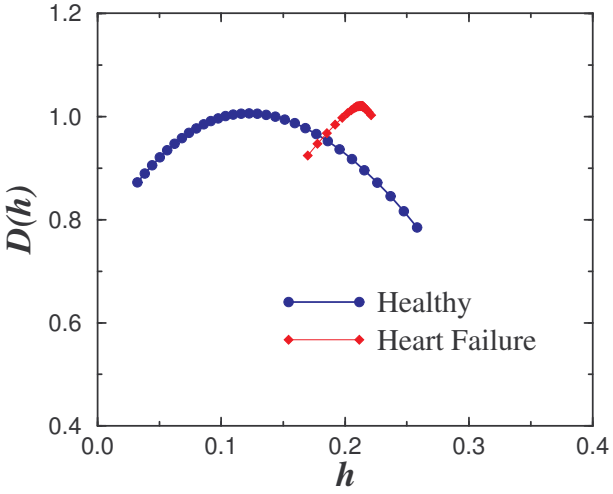

Figure 1: $D(h)$, fractal dimensionality of heartbeat intervals of a healthy heart (large) and an unhealthy heart (small) taken from P.Ch. Ivanov et al. [7]

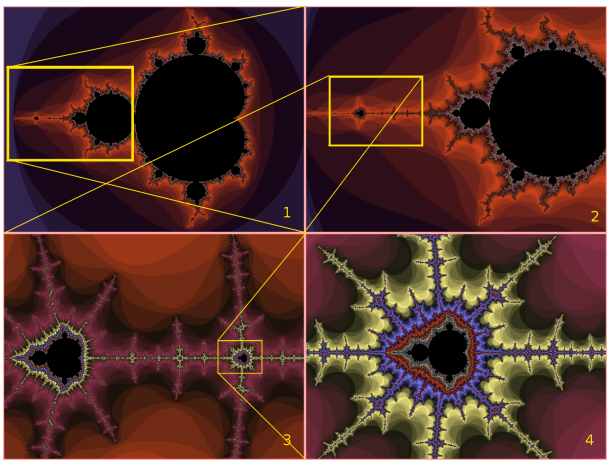

Figure 2: An example of a Mandelbrot fractal, zoomed-in, showing self-similarity at multiple scales.

1.3 Wavelets and Multifractals

We rely on Gauss derivative-based wavelets to detect multifractals [2]. This technique uses wavelets to find regions that are self-similar, it is used to determine important functions such as $\tau(q)$ and $D(h)$ that are explained below.

Wavelets are similar to Fourier transforms. The first difference is that wavelets use a wide variety of functions, such as derivatives of the Gauss function, to compose signals while the Fourier transform tends to focus on sine-waves. The second important difference is that the Fourier transform splits the frequency space into equal sized bins. This means that more bins are allocated to high frequencies than low frequencies. Essentially low frequencies are not given the same amount of representation that medium to high frequencies are given because there are fewer low frequency bins. Wavelets attempt to address this issue by analyzing the spectrum at multiple resolutions and multiple bin sizes. To analyze low frequencies the wavelet transform uses more of the signal per each low frequency bin, while for high fre-

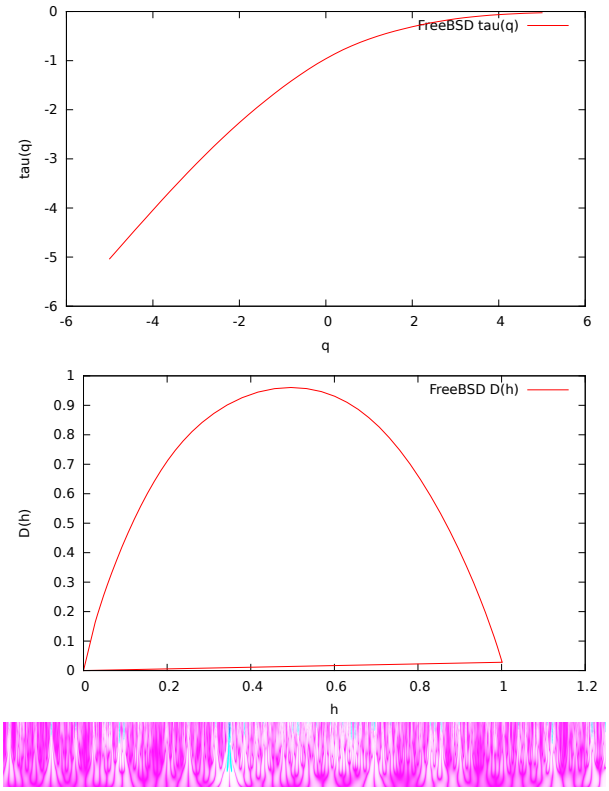

Figure 3: $\tau(q)$, $D(h)$ and Wavelet spectrogram of changes per time unit over the entire lifetime of *FreeBSD*.

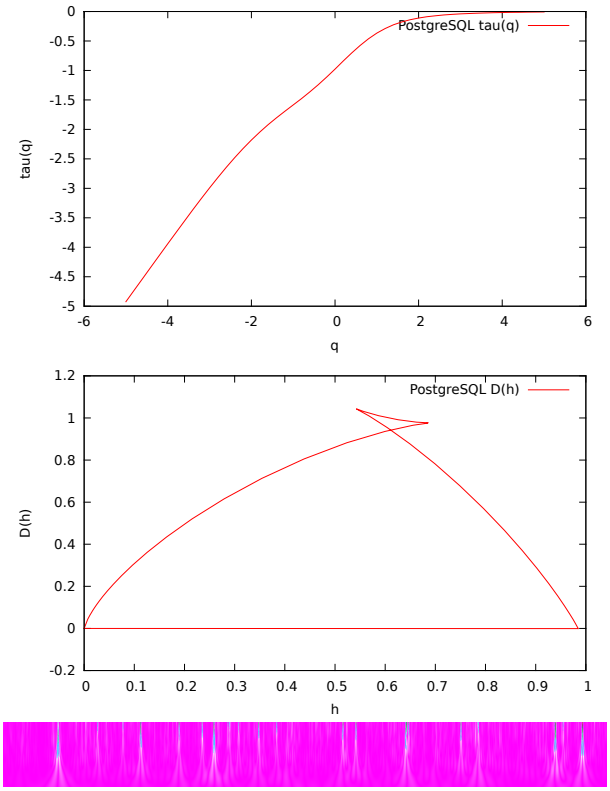

Figure 4: $\tau(q)$, $D(h)$ and Wavelet spectrogram of changes per time unit over the entire lifetime of *PostgreSQL*.

quencies the wavelet transform can use shorter segments of signals. For examples of wavelets see the bottom of Figure 3 and Figure 4.

The next result of this analysis is the partition function. The partition function provides signal partitions at different scales. These partitions divide the signal into self-similar parts at different frequencies. From these partitions, these maxima lines, we can derive the q and $\tau(q)$ values that are related to the scaling and partitions of a signal. The plot of $\tau(q)$ versus q allows us to determine if signal is multifractal. A multifractal signal's $\tau(q)$ function will not have a constant slope, and will not be linear, it will be lumpy with concavities [7].

The next output is $D(h)$, the fractal dimensionality, which demonstrates the distribution of the signal's dimensionality. The h specifically refers to the Hurst exponent, H. Monofractals have one value of h while multifractals have a range of Hurst exponents. The wider the domain and range of the $D(h)$ function the more likely that the signal is multifractal.

In particular in this paper when we refer to dimensionality we mean the fractal dimensionality ($D(h)$). This is a kind of dimensionality representation that includes fractional dimensions. That is dimensionality can be fractional.

2. MULTIFRACTALS IN SOFTWARE DEVELOPMENT

In this section we will demonstrate that the changes per time unit signal of many software projects exhibit multifractal behaviour. This means that change events over time of a software project are self-similar at multiple scales but also that the signals themselves are quite complex and not necessarily captured by many kinds of timeseries analysis.

In order to demonstrate that many software system exhibit this multifractal behaviour we took a selection of *Free/-Libre Open Source Software* (FLOSS) projects hosted by SourceForge, which we had mirrored as of January 22nd 2007, that were part of either the top 250 most popular or most active projects. This resulted in 283 projects with CVS repositories. We also included newer version of Apache 1.3, Boost, Evolution, Firebird, FreeBSD, Gnumeric, PostgreSQL, Samba and SQLite as examples of successful and modern FLOSS projects. In total we had 292 projects.

For this experiment, testing for multifractal properties, we have 3 parameters: the projects, the number of bins (a function of time-unit size), and the N^{th} derivative of the Gauss function, which is the kernel used by our wavelet analysis. The range of Gauss derivatives were from the 0^{th} to the 7^{th} derivative. In terms of bins we take the lifetime of a project and we aggregate it to 1024, 4096 and 8192 bins without smoothing. We found we needed to use a minimum of 1024 bins for the wavelet analysis as we ported the multifractal tool from PhysioNet [4, 2].

For all of our parameters, 292 projects, 3 bin sizes and 8 Gauss derivatives, we had a total of 7008 experiments (24 experiments per project) to determine if a project with said parameters was multifractal.

For each experiment, to determine if the signal is considered multifractal or not, we calculate the wavelet spectrogram with the specified Gauss derivative kernel, we then plot the $\tau(q)$ of the project and determine if it is a straight line or if it has concavity. We also plot the $D(H)$ to determine the range of dimensionality h of the signal. Essentially we threshold the range of the dimensionality of the signal as well as the curvature of the $\tau(q)$ line.

To automate the test of if a signal is multifractal we rely on two sub-tests. Our first sub-test of multifractility is to get the bounding box of $D(H)$ versus h and then we determine the length of the hypotenuse. If this length is greater than 0.1 then the signal might be multifractal. The second sub-test is to measure the maximum second derivative of the $\tau(q)$ versus q, if $|\max \tau''(q)| > 0.01$ then we consider that the signal might be multifractal. If a signal passes either of these sub-tests, we assume that the signal is multifractal.

2.1 Results

We found that with our liberal definition of multifractal properties that for 75% of the projects evaluated, 87% of their 24 tests reported positive multifractal properties. The bottom quarter of projects had ranges of dimensionality (h) less than 0.6137 thus if we rely on the stricter interpretation of multifractal properties extracted from other work [7], we find that the majority of all projects are still multifractal under conservative guidelines.

In terms of the range of Hurst exponents H, 1/4 of all projects had a range of 0.61 or less. 1/2 of all projects had a range of less than 0.97, 3/4 of the projects had ranges less than 1.28 where as the top 1/4 of ranges were between 1.28 and 2.15. This shows that the signals being analyzed have a wide and healthy range of Hurst exponents and fractal dimensionality.

Often the Gauss function (the 0^{th} derivative) would produce negative results as 45% of these tests were negative for multifractal properties. Where as for the 1^{st} to 7^{th} derivatives only 8% (6^{th}) to 12% (1^{st}) of the tests were negative for multifractal properties.

The relationship between the median range of Hurst exponents of the tests run and the number of commits in a project was not linearly correlated (Pearson 0.04) but had a medium rank-based correlation (Spearman 0.50).

2.2 Discussion

Thus we have shown that many FLOSS projects exhibit multifractal development behaviours in terms of changes per time unit. This implies that these change signals are quite complicated and not necessarily easily modeled by time-series analysis techniques that assume or require certain behaviours from the data they analyze.

What these results also confirm is that software development is an inherently self-similar stochastic process. Our current models of agile and iterative development have already recognized the ebb and flow of development via repeating behaviours at many scales from full iterations, to phases, to even story-card implementation. Although this method is not necessarily observing these higher level behaviours we wonder if this is a component of the multi-scale self similarity that is being exhibited by these processes.

2.3 Future of Multifractals in Software Engineering

We plan to take this research further and investigate periods of development when the dimensionality of a signal collapses. This is analogous to the heartbeat modeling research [7] where they found that the heartbeat intervals of healthy and dying hearts looked very similar from a time-series perspective but when multifractal analysis was applied it was noticed that the dimensionality of an dying heart had collapsed, to a smaller range than the healthy heart. Currently we are building a case study to explain why the dimensionality of a changes per time unit signal collapses during certain periods.

We are also investigating the multifractility of call depth and stack depth during testing. Preliminary investigation reveals that some dynamic traces exhibit multifractal behaviours while others do not. We seek to investigate the significance of this behaviour. One use of multifractal analysis is that we can partition development by self-similarity at different scales.

3. CONCLUSIONS

In this paper we have demonstrated some initial results. We showed that the changes over time in many of the 292 FLOSS projects' development history exhibit multifractal properties of self-similarity and multiple fractal dimensions. We demonstrated their change per time unit behaviour is multifractal.

Multifractal properties are important because these explain the complexity of a signal, and why many models, such as ARIMA, fall short. Essentially, development behaviour resembles physiological behaviour.

4. REFERENCES

[1] *NIST/SEMATECH e-Handbook of Statistical Methods*, 2008. http://www.itl.nist.gov/div898/handbook/.

[2] Y. Ashkenazy. Software for analysis of multifractal time series. http://www.physionet.org/physiotools/multifractal/, November 2004.

[3] M. D'Ambros, M. Lanza, and H. Gall. Fractal figures: Visualizing development effort for cvs entities. In *VISSOFT*, pages 46–51. IEEE Computer Society, 2005.

[4] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13).

[5] I. Herraiz, J. M. Gonzalez-Barahona, and G. Robles. Forecasting the number of changes in eclipse using time series analysis. In *MSR '07*, page 32, Washington, DC, USA, 2007. IEEE Computer Society.

[6] I. Herraiz, J. M. Gonzalez-Barahona, and G. Robles. Towards a theoretical model for software growth. In *MSR '07*, page 21, Washington, DC, USA, 2007. IEEE Computer Society.

[7] P.Ch. Ivanov, L. A. N. Amaral, A. L. Goldberger, S. Havlin, Rosenblum, Z. R. Struzik, and H. E. Stanley. Multifractality in human heartbeat dynamics. *Nature*, 399:461–465, June 1999.

[8] J. Wu. *Open source software evolution and its dynamics*. PhD thesis, Waterloo, Ont., Canada, Canada, 2006. AAINR14637.