Presentation at CiE2012, Cambridge, 19 June, 2012.

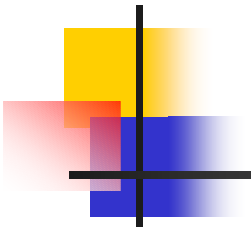
“Equilibriation of information in software systems"

Les Hatton

www.leshatton.org

Version 1.2: 17/June/2012

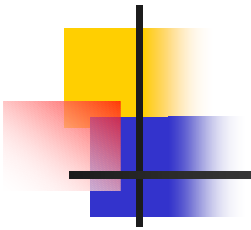
Overview



- A little about software defects
- A hidden clockwork
- Conclusions

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

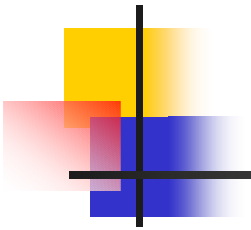
So what can we say about defect ?



- Good
  - Computer scientists have researched the average *density* of defect in code extensively, (0.1 – 10 defects /KSLOC)
- Not so good
  - Little progress in quantifying the *effects* of such defect on numerical results
  - Lots of empirically fitted models but no underlying theory for defect density, so we don't know how well something has been tested. (There are two ways of having low defect density !)

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

Overview

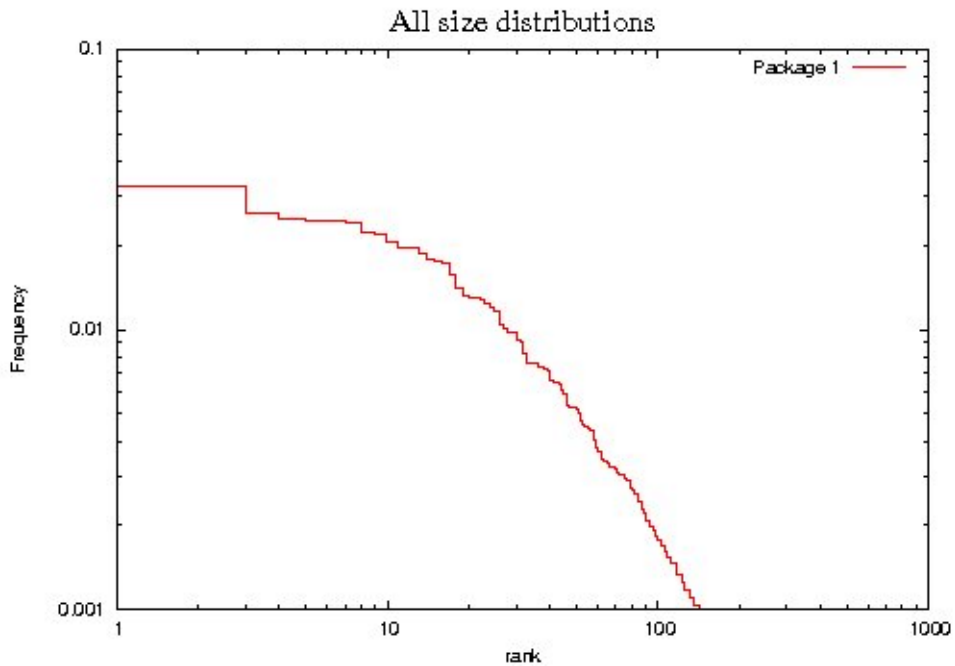


- A little about software defects
- A hidden clockwork
- Conclusions

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

Software size distributions appear power-law in LOC

In spite of this, systems appear astonishingly similar in their information properties …



Smoothed (cdf) data for 21 systems, C, Tcl/Tk and Fortran, combining 603,559 lines of code distributed across 6,803 components, (Hatton 2009, IEEE TSE)

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

Searching for a hidden clockwork: building systems

- When we build a system we are making choices
  - Choices on functionality
  - Choices on architecture
  - Choices on programming language(s)
- There is a general theory of choice – Shannon information theory.
- If we constrain both choice and size in any symbol-based system and use the formalism of statistical mechanics, we get …

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

The alphabetic power-law theorem

$$p_i \sim (a_i)^{-\beta}$$

This states that in any software system, conservation of size and information (i.e. choice), it is overwhelmingly likely that the probability of a component appearing with a total of t_i tokens obeys a power-law in a_i which is the list of <u>unique</u> tokens used to build the i^{th} component. Tokens are either fixed by the language, (if, then, else ...) or can be created by the programmer, (identifiers, constants ...).

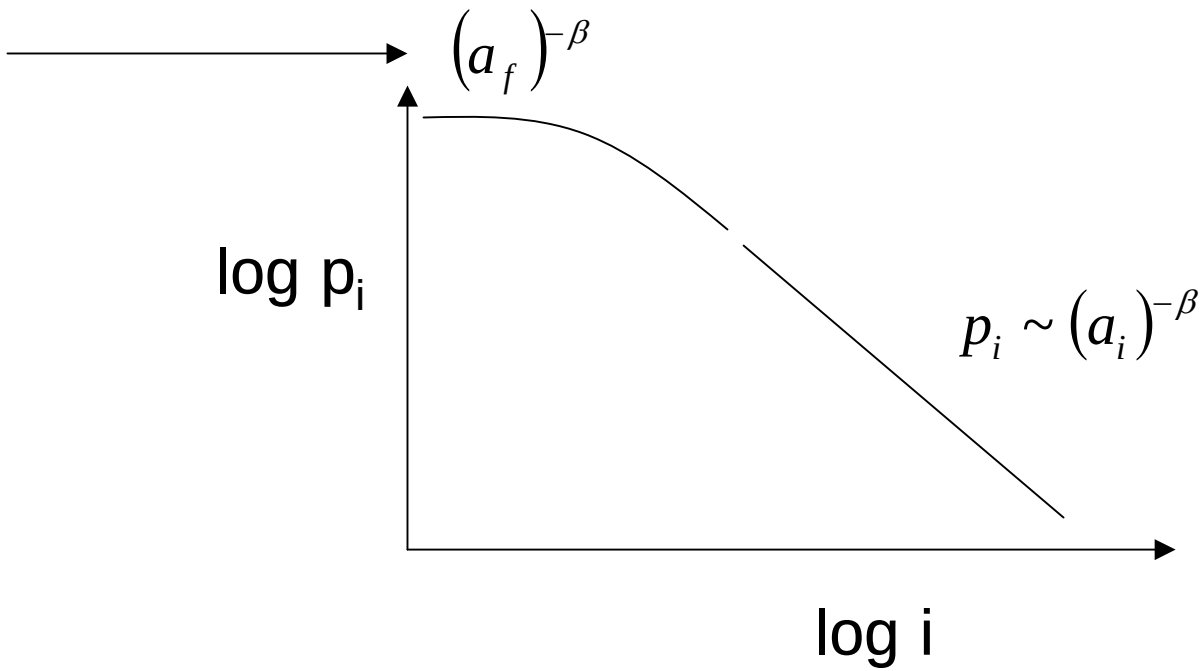
Hatton L (2011) IFIP, Boulder Colorado August 2011.

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

Application to software systems

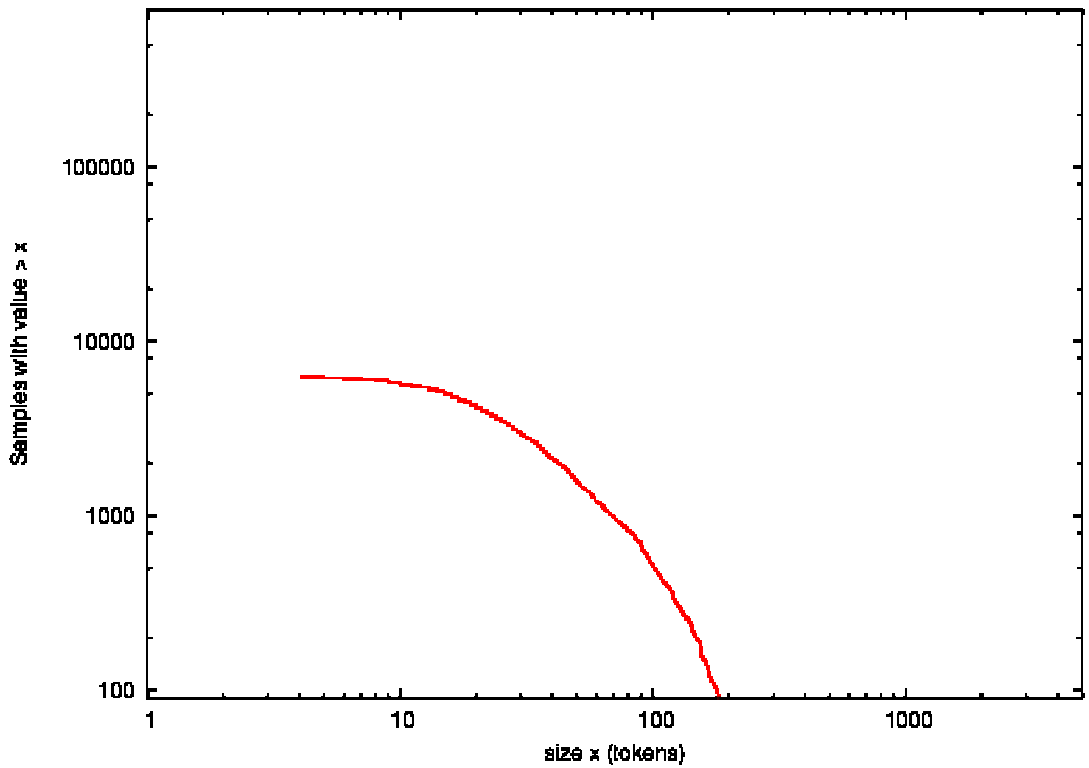
We can test this. We are looking for this signature ...

Fixed symbols of the language used. *Small programs are dominated by the fixed tokens of a programming language.*



Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

Equilibriation in 400,000 line chunks



42 million lines of Ada, C, C++, Fortran, Java, Tcl-Tk from 80+ systems

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

So, what can we say about defects ?

- However, systems evolve such that the total number of defects D is conserved, (since they are finite), (Hatton, (2009) IEEE TSE, 35(4), p. 566-572), giving

$$D = \sum_{i=1}^{M} d_i \Rightarrow p_i = \frac{1}{Q(\gamma)} e^{-\gamma \frac{d_i}{t_i}}$$

- Combining this with

$$p_i \sim (a_i)^{-\beta}$$

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

We get the tloga theorem

- Predicts the following defect distribution

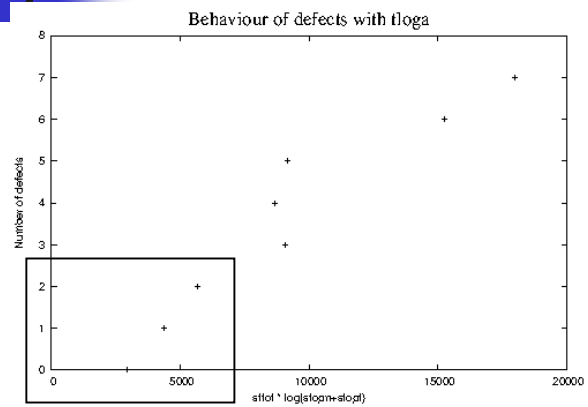
$$d_i \approx t_i \log a_i$$

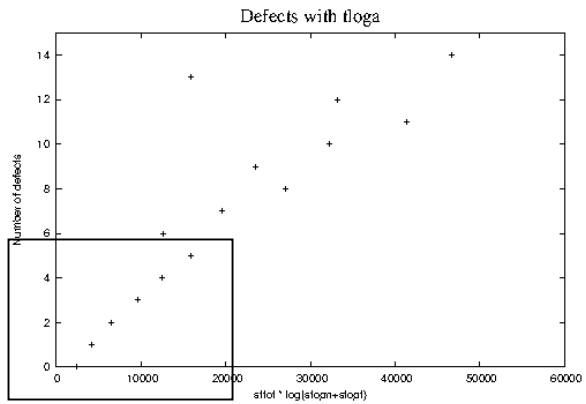
Here t_i is the total number of keywords and identifiers and d_i is the number of defects.

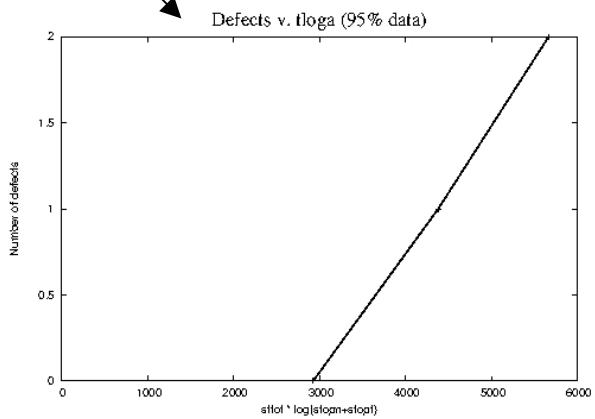
- Can we test this ?

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

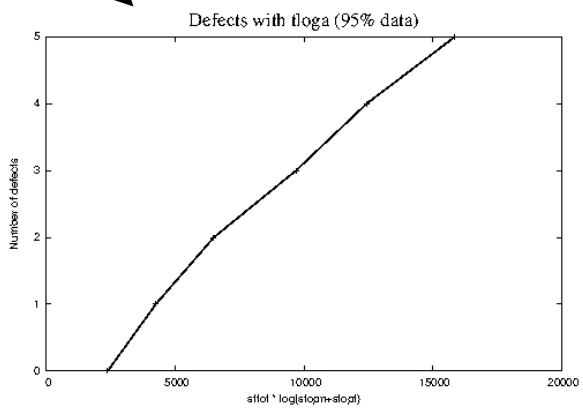
The tloga theorem







95%

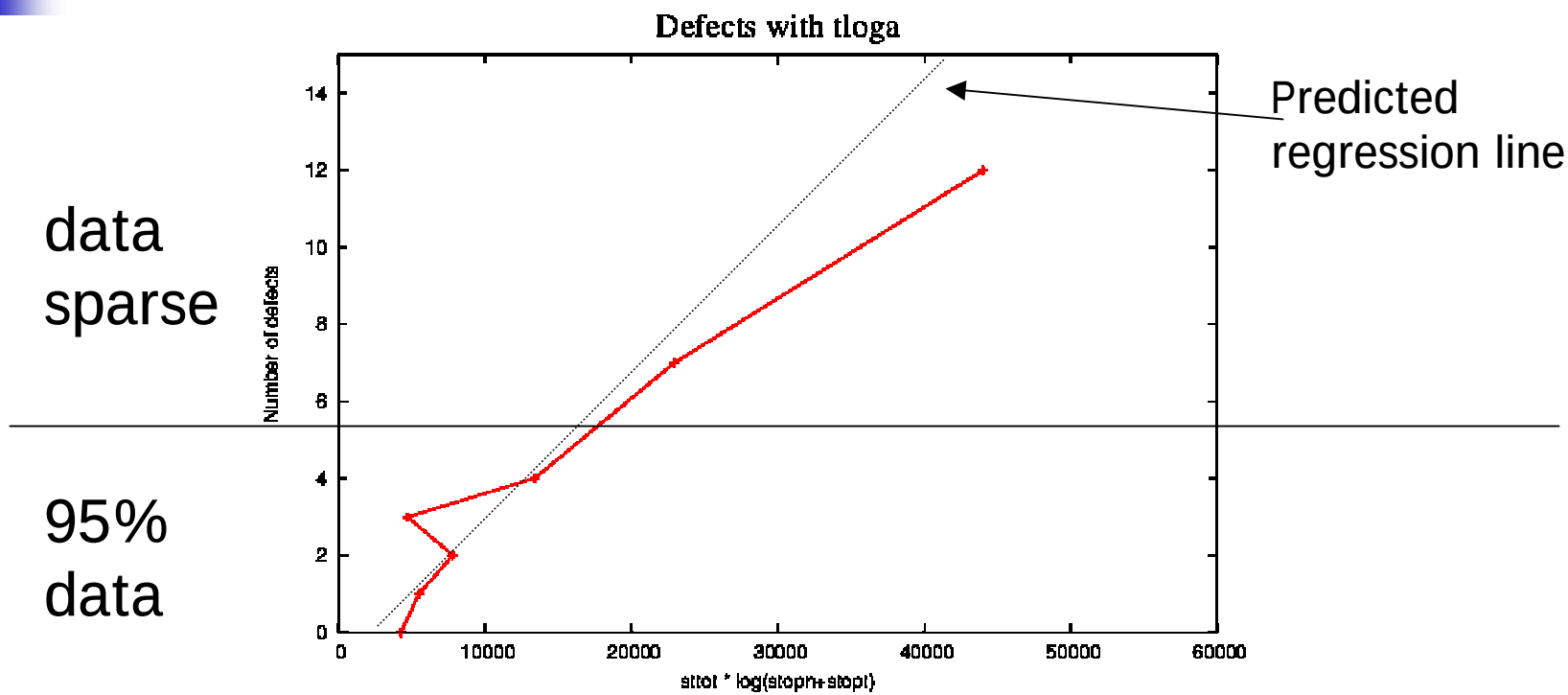


NAG Fortran Library

Eclipse IDE (Java)

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

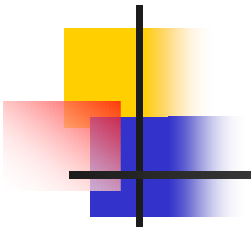
Equilibriation to tloga in the Eclipse IDE*



*With grateful thanks to Andreas Zeller et. al. (2007) for extracting the defect data and making it openly available. http://www.st.cs.uni-sb.de/softevo. The data comes from releases 2.0,2.1 and 3.0. There are 10,613 components in the release 3.0.

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

Overview



- A little about software defects
- A hidden clockwork
- Conclusions

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

Conclusions

$$p_i \sim (a_i)^{-\beta} \qquad d_i \sim t_i \log a_i$$

- Conservation of size and information appear sufficient to force software systems into a power-law unique token distribution, independently of what they do, what they are written in or who wrote them. This is now supported empirically at very high significance.
- Simultaneous conservation of defect appears to give empirically supported asymptotic tloga behaviour in software systems. This should throw light on how well-tested a software system is.

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement

References

My writing site:-

http://www.leshatton.org/

Specifically,

http://www.leshatton.org/variations_2010.html

For comments on reproducibility in software systems,

http://www.nature.com/nature/journal/v482/n7386/full/nature10836.html

Thanks for your attention.

Copyright Les Hatton, 2012-. Copying freely permitted with acknowledgement