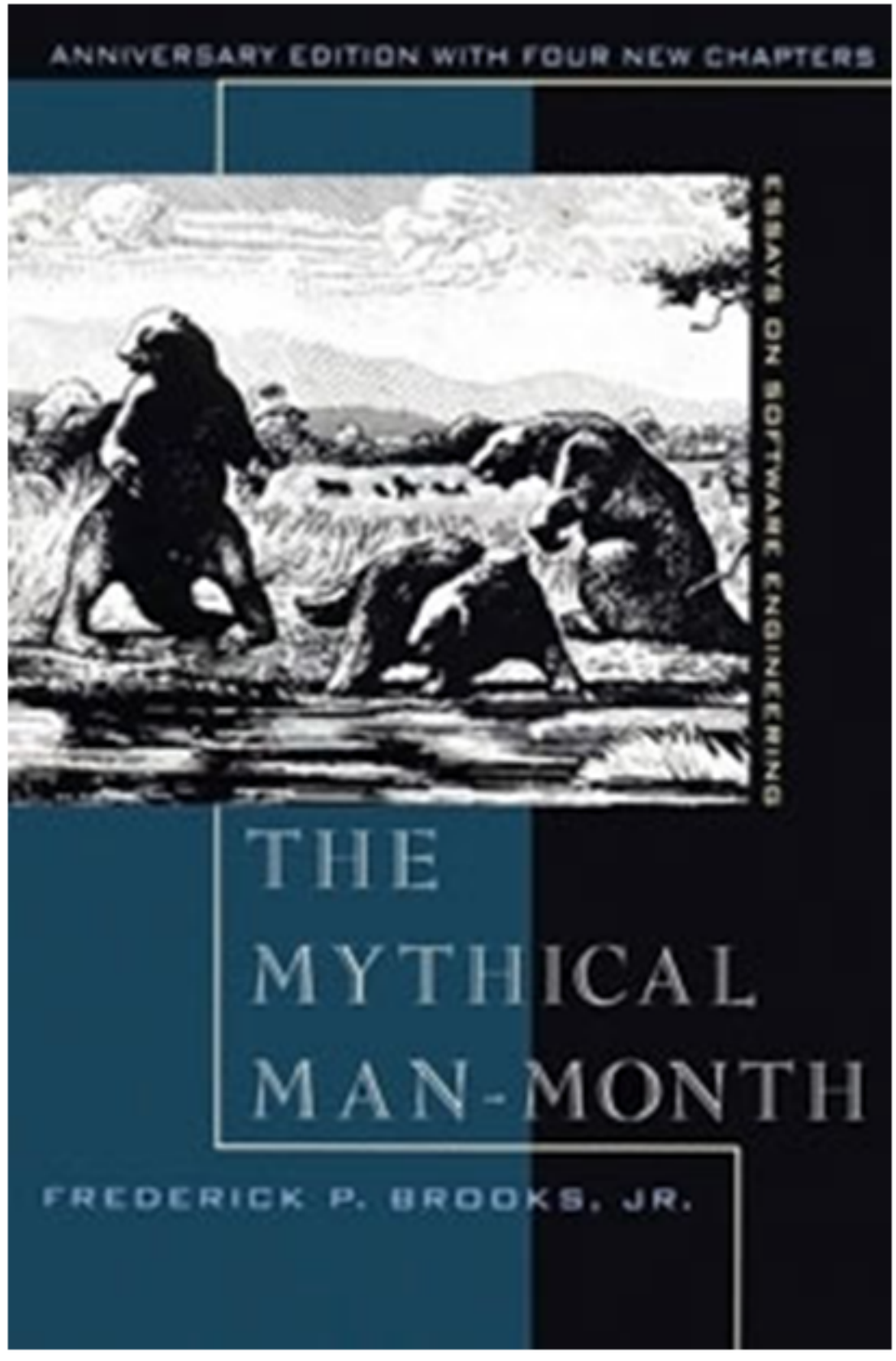
SE4831 Software Quality Assurance
Pareto Principle

- Objectives
  - (a) Explain how the pareto principle can be used during software development.
  - (b) Explain how institutional data and the pareto principle can be used to yield better inspection performance.


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
ESSAYS ON SOFTWARE ENGINEERING
THE
MYTHICAL
MAN-MONTH
FREDERICK P. BROOKS, JR.

10. Many software phenomena follow a Pareto distribution: 80% of the contribution comes from 20% of the contributors. Knowing this can help a project focus on the 20% subset providing 80% of the leverage for improvement. Some examples:

- 20% of the modules contribute 80% of the cost;
- 20% of the modules contribute 80% of the errors (not necessarily the same ones);
- 20% of the errors consume 80% of the cost-to fix;
- 20% of the modules consume 80% of the execution time;
- 20% of the tools experience 80% of the tool usage.

To summarize, I think it has been a strong credit to the software metrics field that it has been able to determine and corroborate these and many other useful software metric relationships. And there are many useful new ones coming along. I look forward to reading about them in this column.

INDUSTRIAL SOFTWARE METRICS: A TOP-TEN LIST

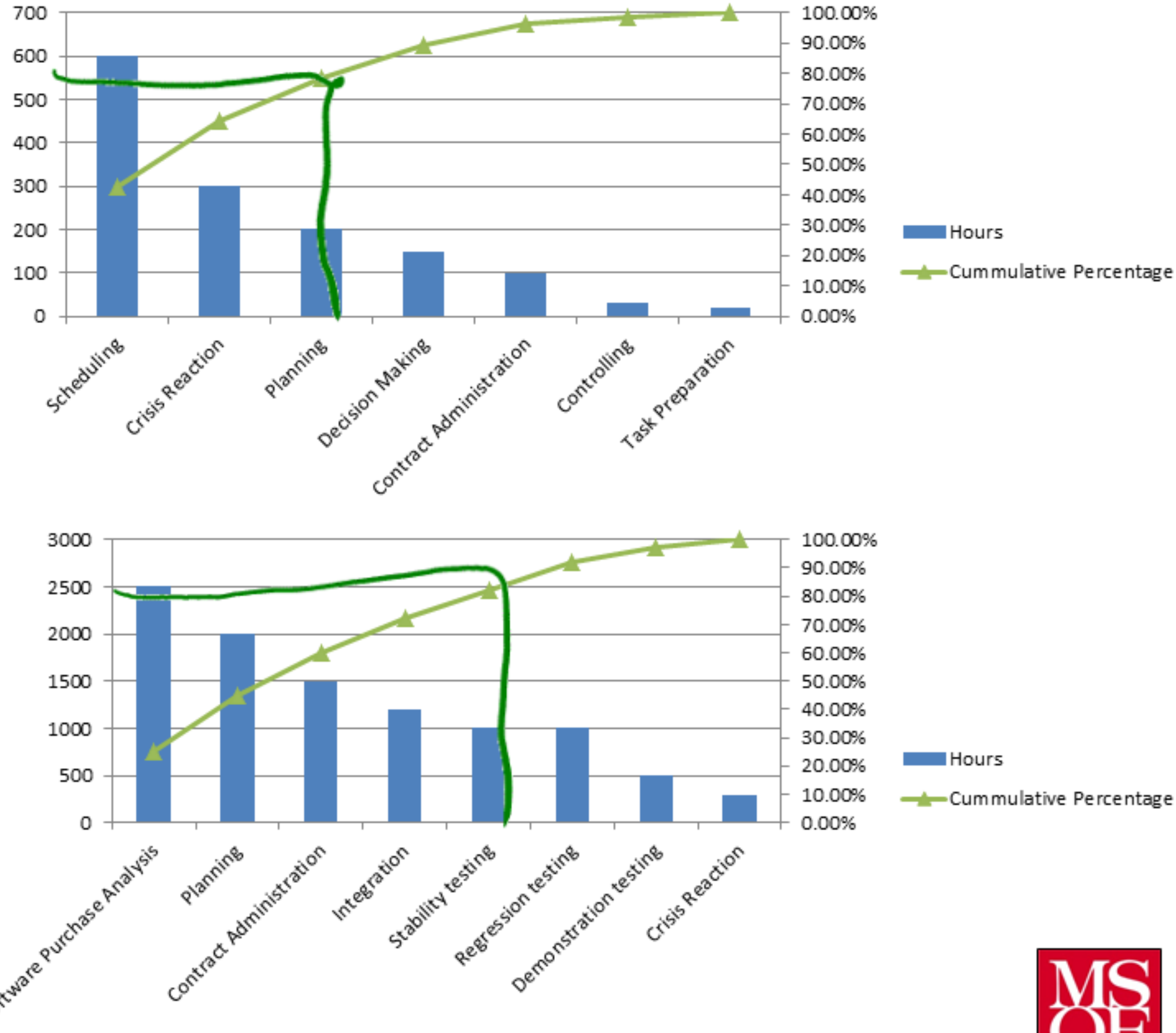
Barry W. Boehm, TRW Inc.

1977 World Wide Military Command and Control System

Personnel Tasks	*Hours Expanded*	*% of Hours Expanded*	*Cumulative Hours Expanded*	*Cumulative % of Hours Expanded*
*Managerial Personnel**				
Scheduling	600	43	600	43
"Crisis reaction"	300	21	900	64
Planning	200	14	1,100	78
Decision making	150	11	1,250	89
Contract administration	100	7	1,350	96
Controlling	30	3	1,380	99
"Task preparation"	20	1	1,400	100
*Technical Personnel**				
Software purchase analysis	2,500	25	2,500	25
Planning	2,000	20	4,500	45
Contract administration	1,500	15	6,000	60
Integration	1,200	12	7,200	72
Stability testing	1,000	10	8,200	82
Regression testing	1,000	10	9,200	92
Demonstration tests	500	5	9,700	97
"Crisis reaction"	300	3	10,000	100

*Assumes 3 management and 22 technical.

WWMCCS Pareto Charts

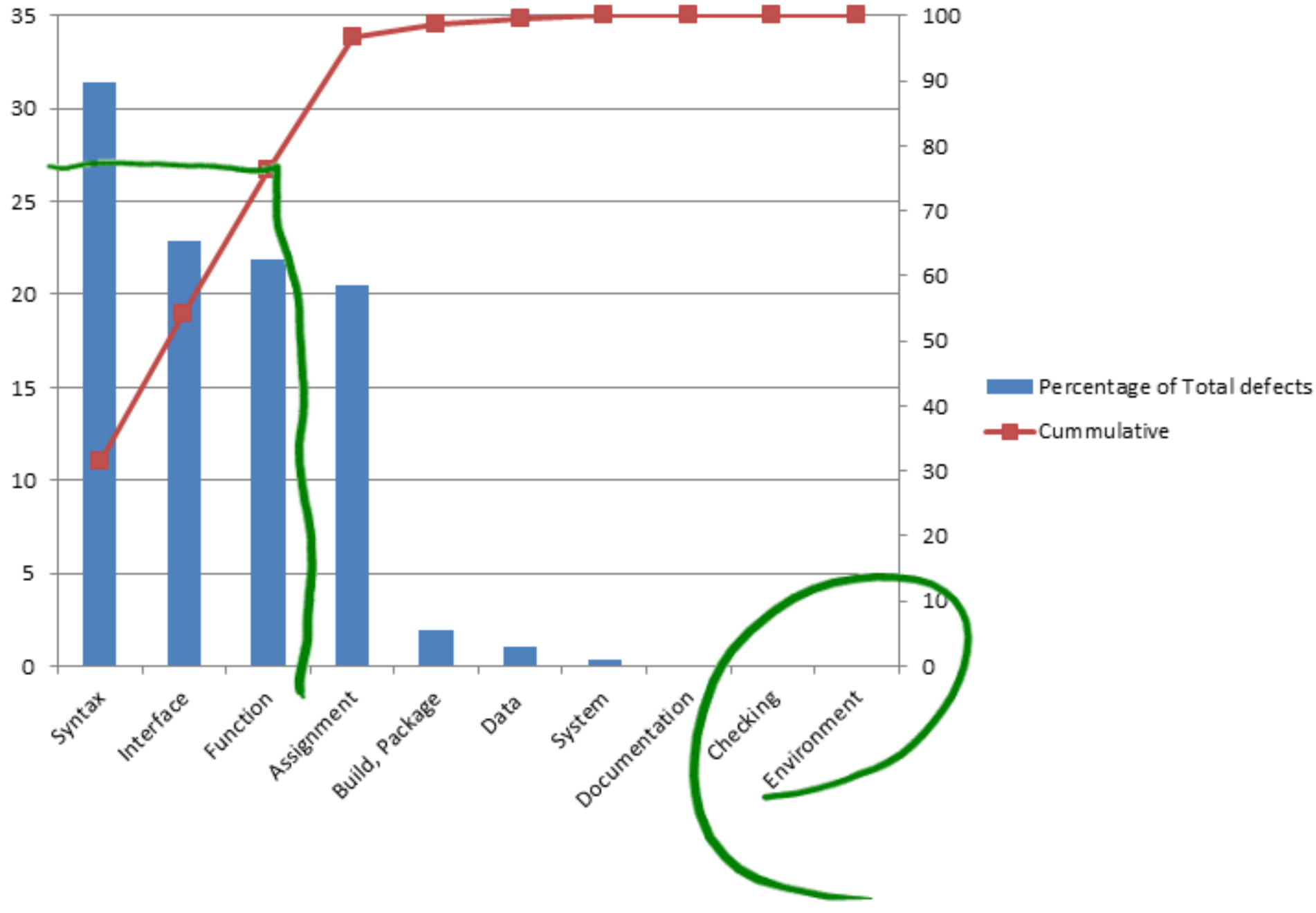


A Pareto Distribution Example from Software

(Humphrey, 1995)

Defect Type	Percentage of Total defects
Documentation	0
Syntax	31.4
Build, Package	1.9
Assignment	20.5
Interface	22.9
Checking	0
Data	1
Function	21.9
System	0.5
Environment	0

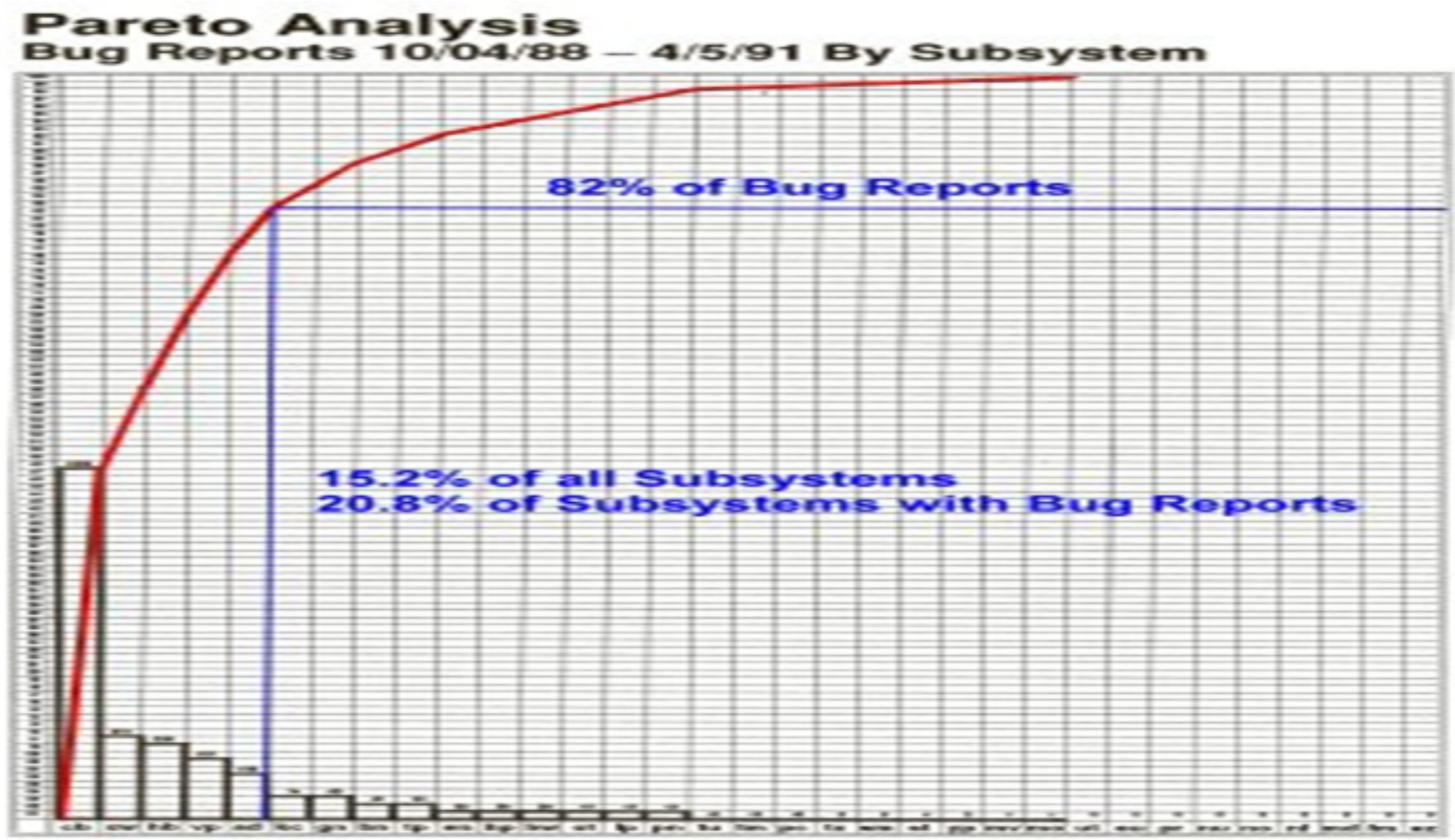
A Pareto Chart



Constructing a Pareto Chart

- Purpose
  - Rank data in priority order
- Conditions
  - The data should belong to the same general class
- Inputs
  - A listing of the data items and pertinent parameters
- Steps
  - Select the parameter to be used as the sorting key
  - Count the number of items in each category
  - Calculate the percentage of total items in that category
  - Sort the categories in descending order by frequency of occurrence
  - Plot sorted data
- Interpretation
  - Items farthest to the left should receive the most attention

A Real World Example

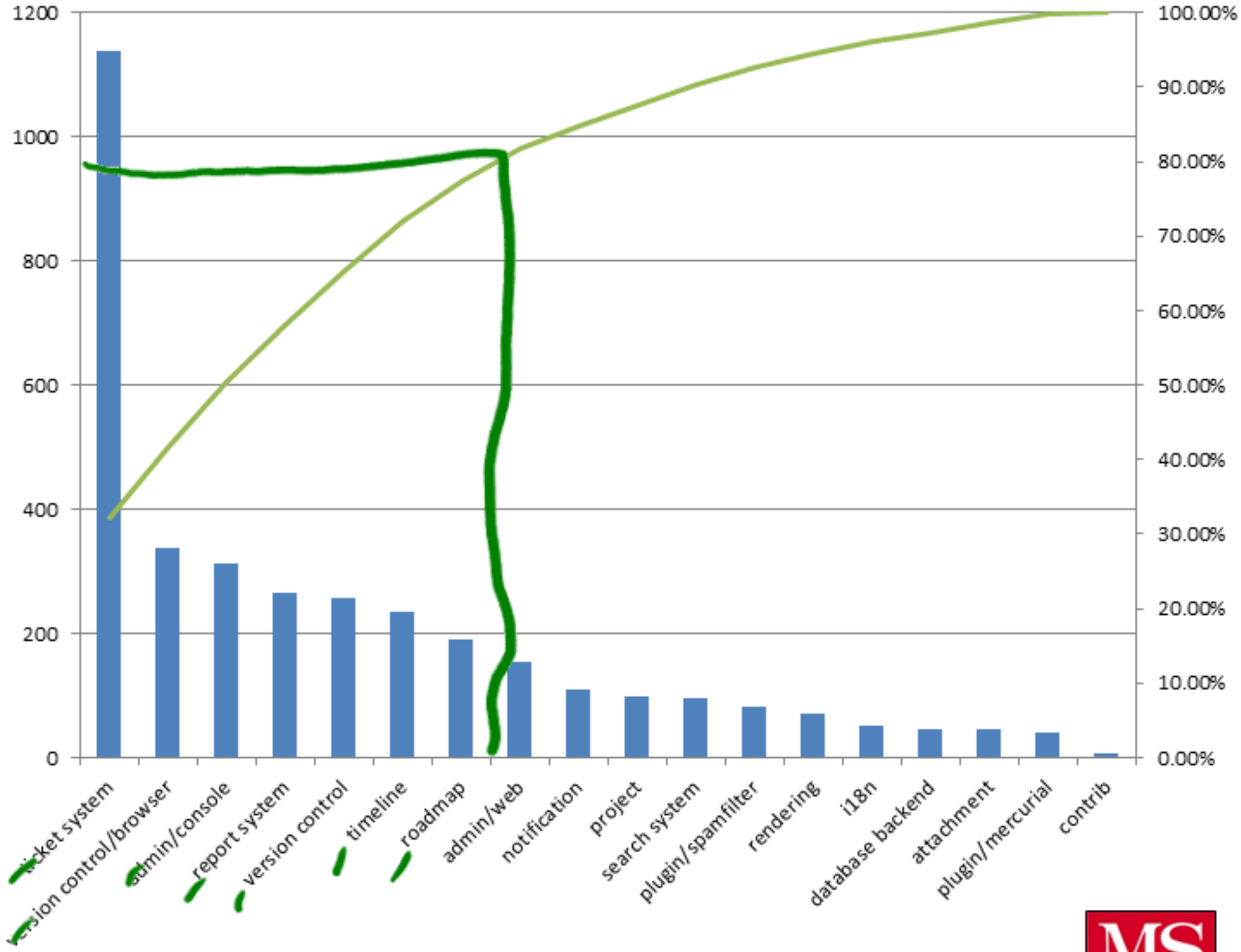


- 82% of the bug reports were from 15.2% of all subsystems
  - 20.8% of the subsystems with actual bug reports, fitting closely to the Pareto
- Similar results were revealed when the first subsystem was eliminated:
  - 82% of the bug reports applied to about 21% of all the remaining subsystems.

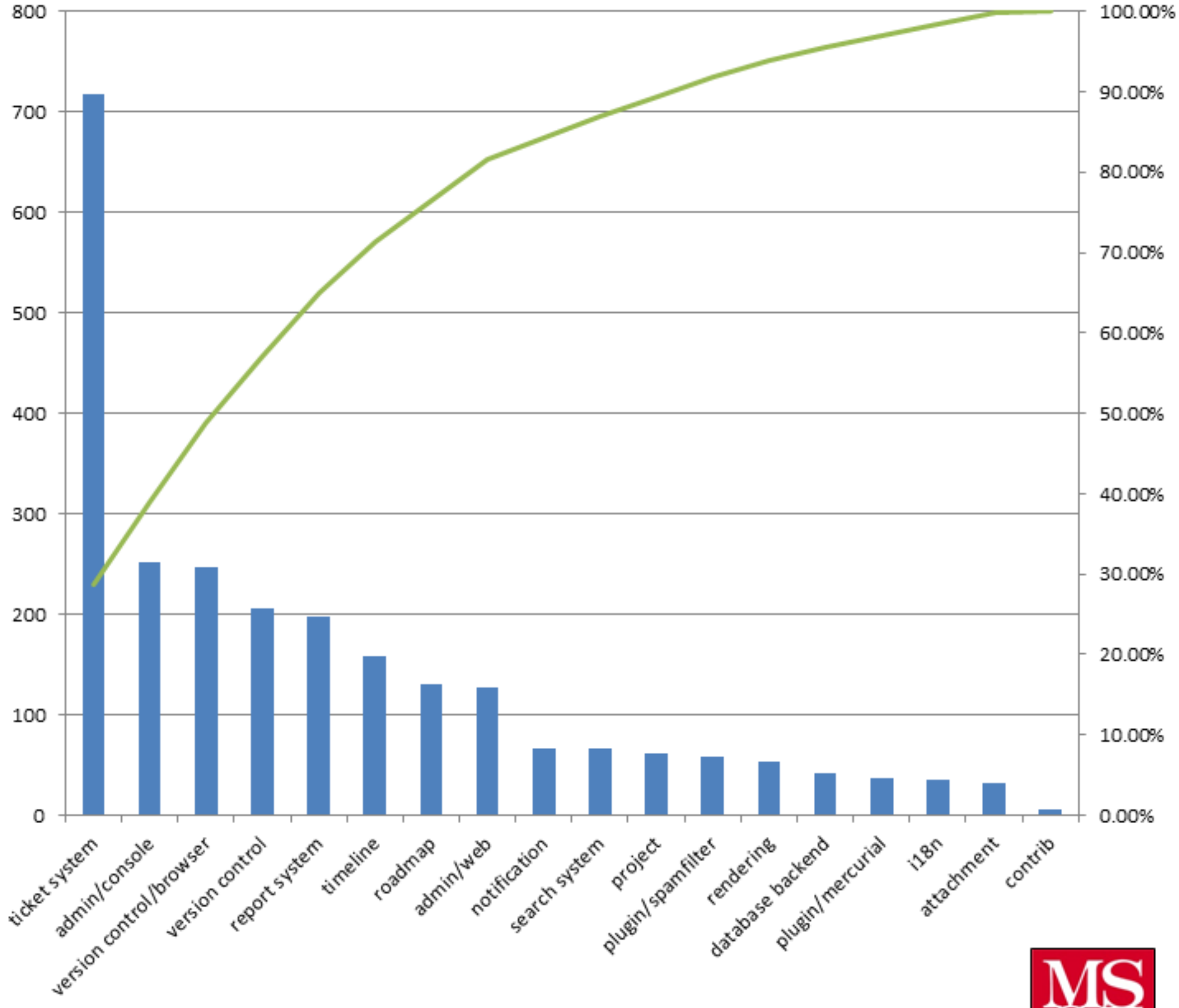
Another real world example

- 3535 Bug reports entered in the system
  - Spread across 17 subsystems
  - Where are the problems at?

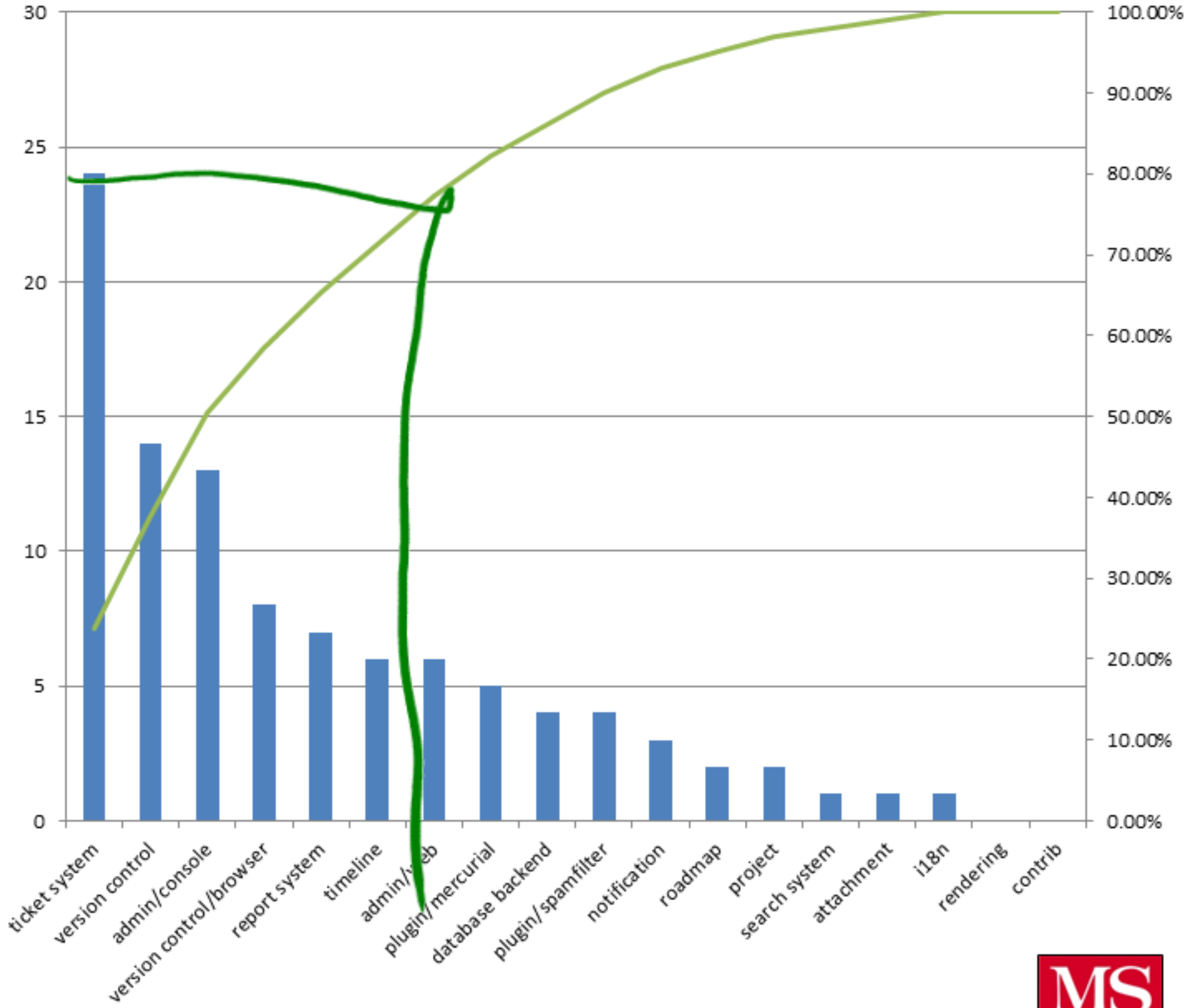
Raw Analysis based on component



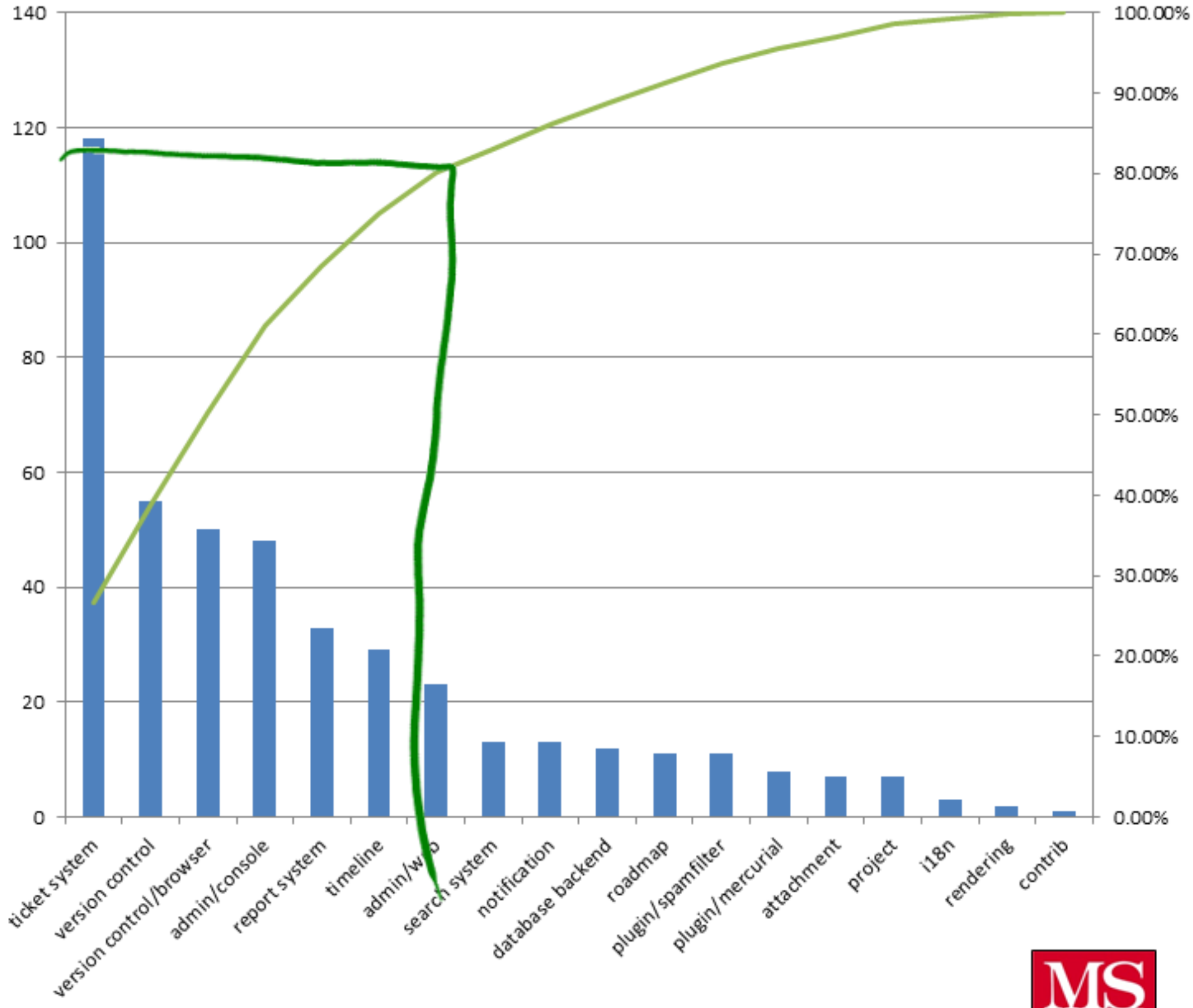
Raw Analysis, Just Defects



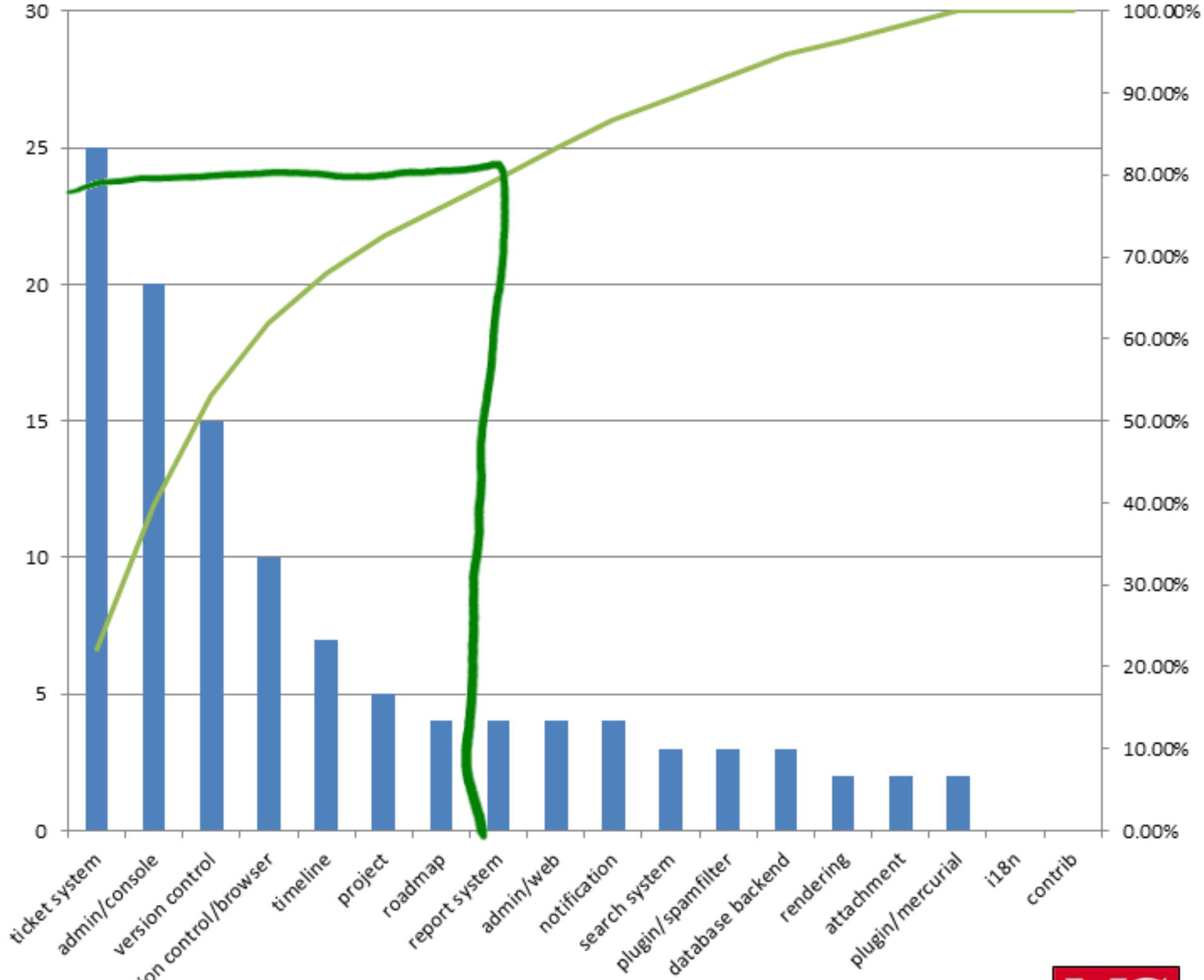
Critical Defects



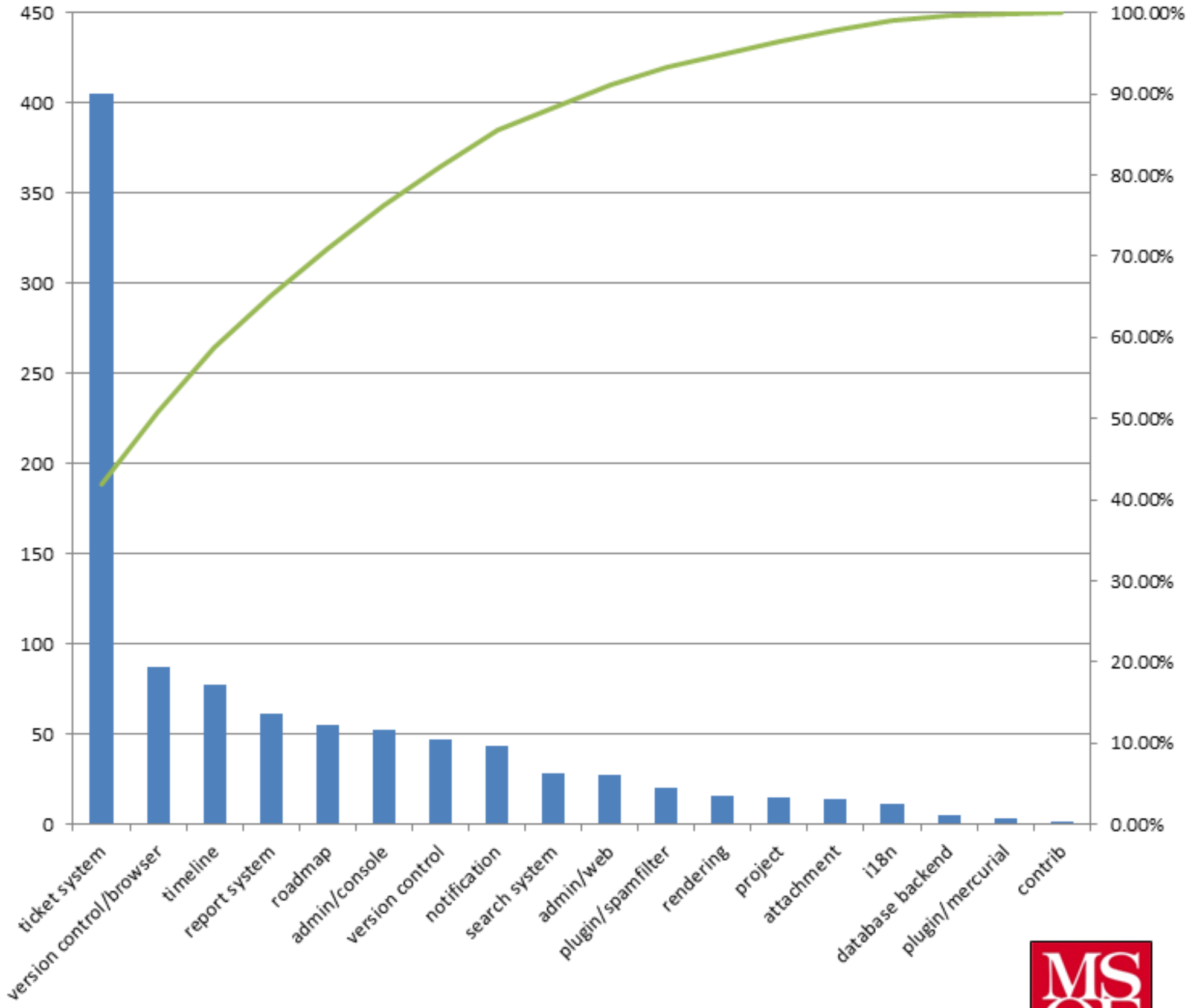
Most Severe Defects
(Larger Set than Critical)



Highest Priority Defects



Enhancements



Sort Orders

Raw Analysis	Just defects	Critical Defects	Most severe defects	Highest Priority defects	Normal to Highest Priority Defects	Enhancements
ticket system	ticket system	ticket system	ticket system	ticket system	ticket system	ticket system
version control/browser	admin/console	version control	version control	admin/console	version control/browser	version control/browser
admin/console	version control/browser	admin/console	version control/browser	version control	admin/console	timeline
report system	version control	version control/browser	admin/console	version control/browser	version control	report system
version control	report system	report system	report system	timeline	report system	roadmap
timeline	timeline	timeline	timeline	project	timeline	admin/console
roadmap	roadmap	admin/web	admin/web	roadmap	roadmap	version control
admin/web	admin/web	plugin/mercurial	search system	report system	admin/web	notification
notification	notification	database backend	notification	admin/web	notification	search system
project	search system	plugin/spamfilter	database backend	notification	project	admin/web
search system	project	notification	roadmap	search system	search system	plugin/spamfilter
plugin/spamfilter	plugin/spamfilter	roadmap	plugin/spamfilter	plugin/spamfilter	plugin/spamfilter	rendering
rendering	rendering	project	plugin/mercurial	database backend	rendering	project
i18n	database backend	search system	attachment	rendering	database backend	attachment
database backend	plugin/mercurial	attachment	project	attachment	i18n	i18n
attachment	i18n	i18n	i18n	plugin/mercurial	plugin/mercurial	database backend
plugin/mercurial	attachment	rendering	rendering	i18n	attachment	plugin/mercurial
contrib	contrib	contrib	contrib	contrib	contrib	contrib