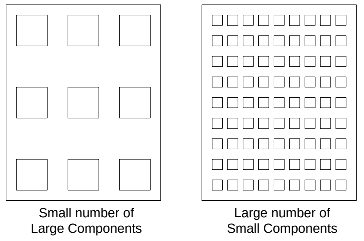
Aggregation Levels in Composable Architectures

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

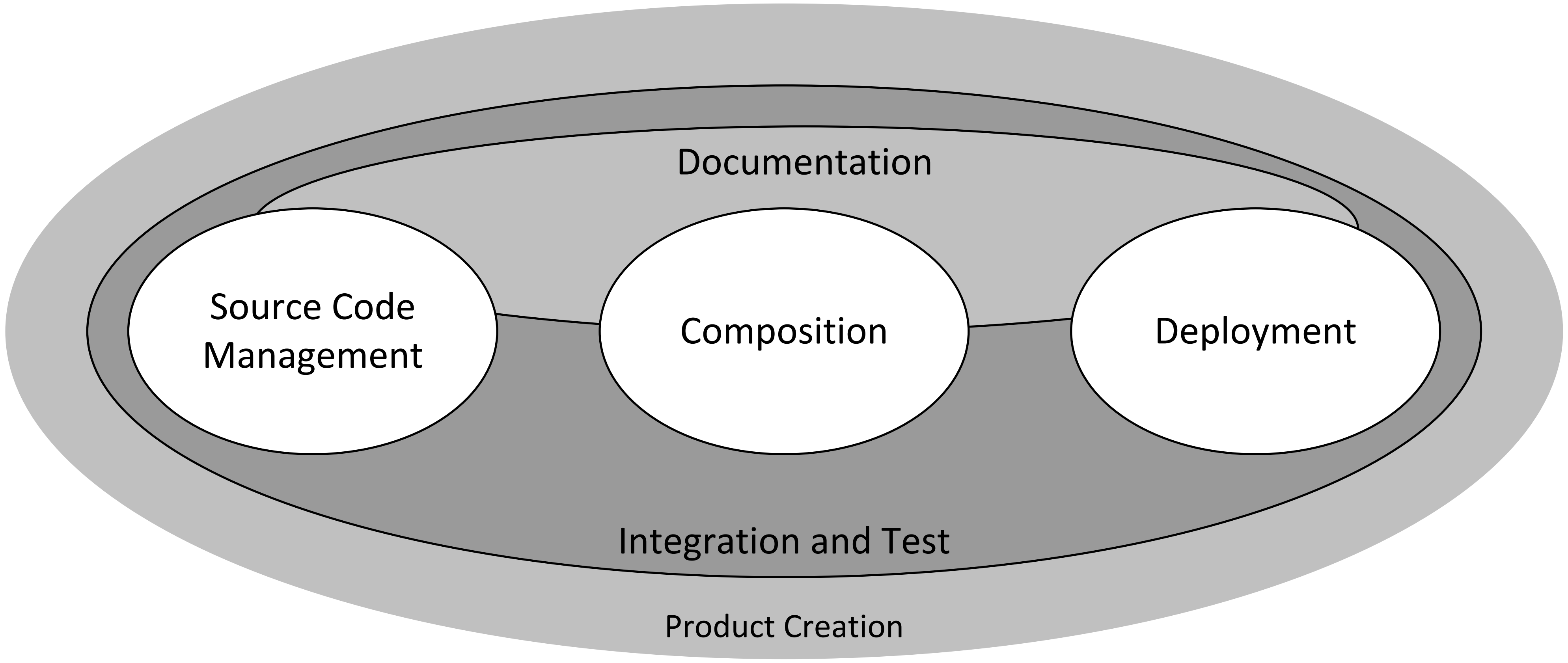
Abstract

The creation of a Product Family is an alternation of decomposition and synthesis steps. The products and intermediate compositions can be viewed as recursive aggregation levels. Careful trade-offs are required between the size of an aggregation level and the way it will be deployed, to balance amongst others flexibility and (configuration) manageability.

September 1, 2020
status: draft
version: 2.4



Aggregation Levels viewpoints



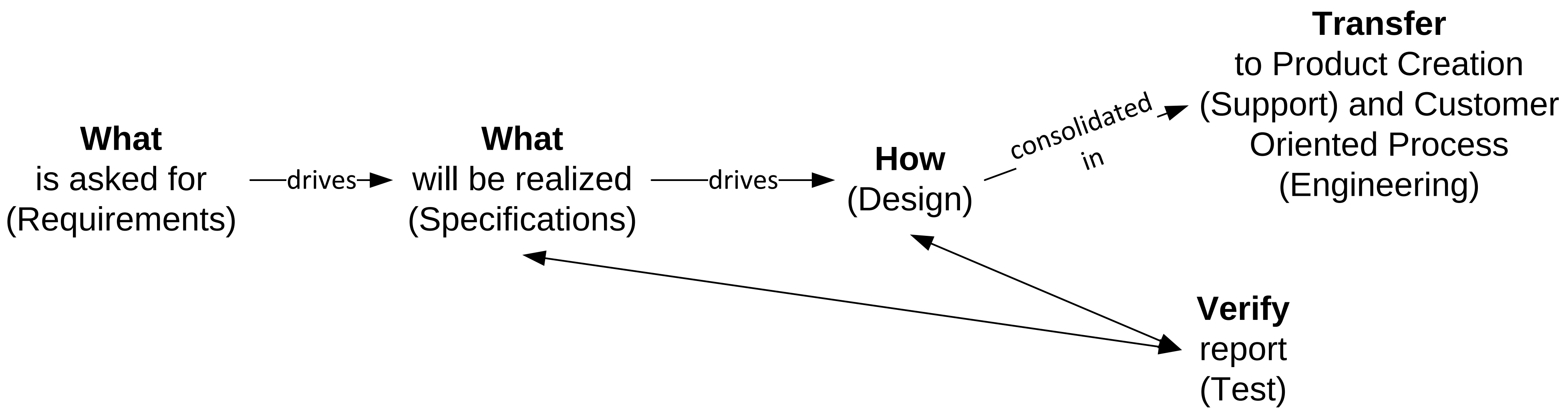
Concerns per viewpoint

Viewpoint	Concerns
Documentation	Requirements, Specification, Design, Transfer, Test, Support
Source Code Management	Storage, Management, Generation
Composition	System, Subsystem, Function, Application
Deployment	Releasing, Distribution, Protection, Update, Installation, Configuration
Integration and Test	Confidence, Problem Tracking

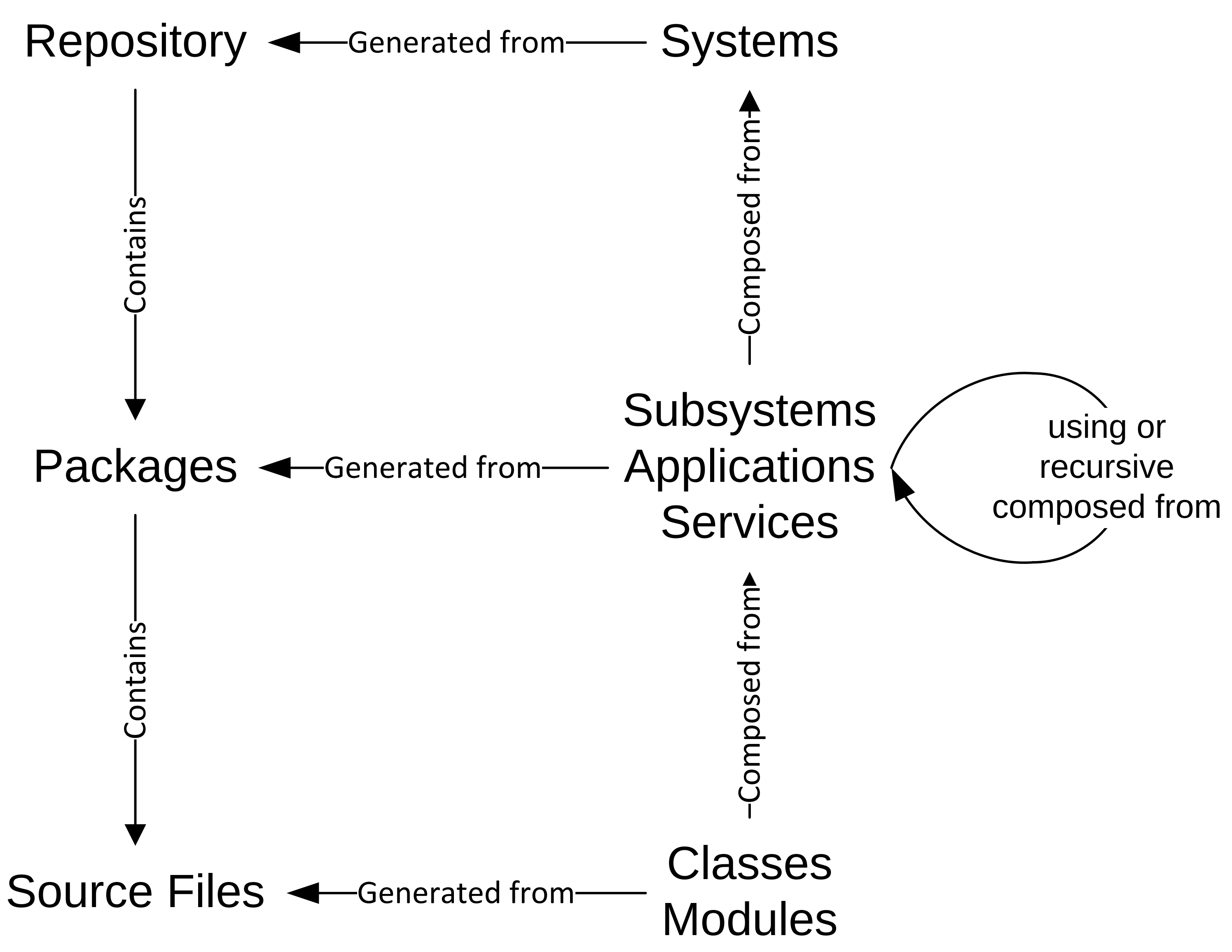
Aggregation Levels or Entities per viewpoint

Viewpoint	Entities
Documentation	Product Family, Product/System, Function/Feature, Subsystem, Component, Building Block, Module
Source Code Management	Package, File
Composition	Product, Executable, Dynamic Library, Component
Deployment	Distribution Medium (”CD”), Unit of Licensing (”SW key”), Package, Patch, Configuration data
Integration and Test	Test Configurations, Intermediate Integration results

Documentation Viewpoint



Repository Viewpoint



Typical Sizes of SW for Aggregation Levels

Entity	Typical size loc	packages
repository	1M-10M	10-100
package	10k–100k	
file	100-1k	

Rules of thumb file-size

- Files should be larger than 100 loc;

  The overhead per file and the ”value” per file must be balanced.

- Files should be less than 1000 loc;

  Large files reduce the overview within the module. Larger files are an indication for a lack of modularity.

Package Size Considerations

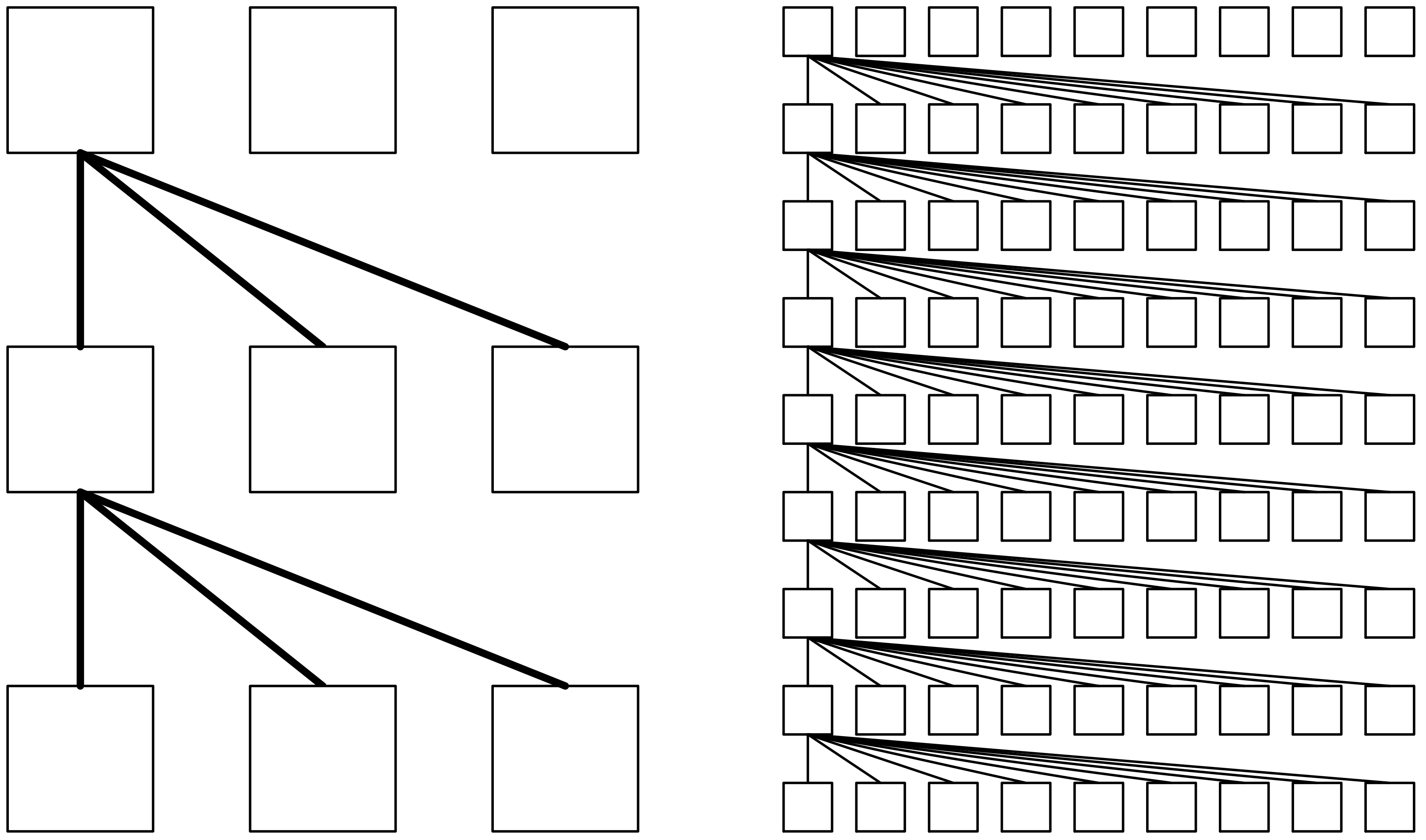
- at least 10 files per package;

  Packaging files or modules generates some overhead in usage and management. The value of this packaging must be substantial to offset this additional overhead.

- at most 100 kloc per package to maintain overview;

  For unambiguous package-ownership and sufficient overview.

Composition Viewpoint: Granularity



Small number of
Large Components

Large number of
Small Components

Nr Components vs Nr of Architects; Naive

Capacity of architects c		10	20	40
Number of components n	Number of relations $r = n\sqrt{n}$	Number of Architects $a = r/c$		
2	3	0	0	0
4	8	1	0	0
10	32	3	2	1
20	89	9	4	2
40	253	25	13	6
100	1000	100	50	25
300	5196	520	260	130
1000	31623	3162	1581	791

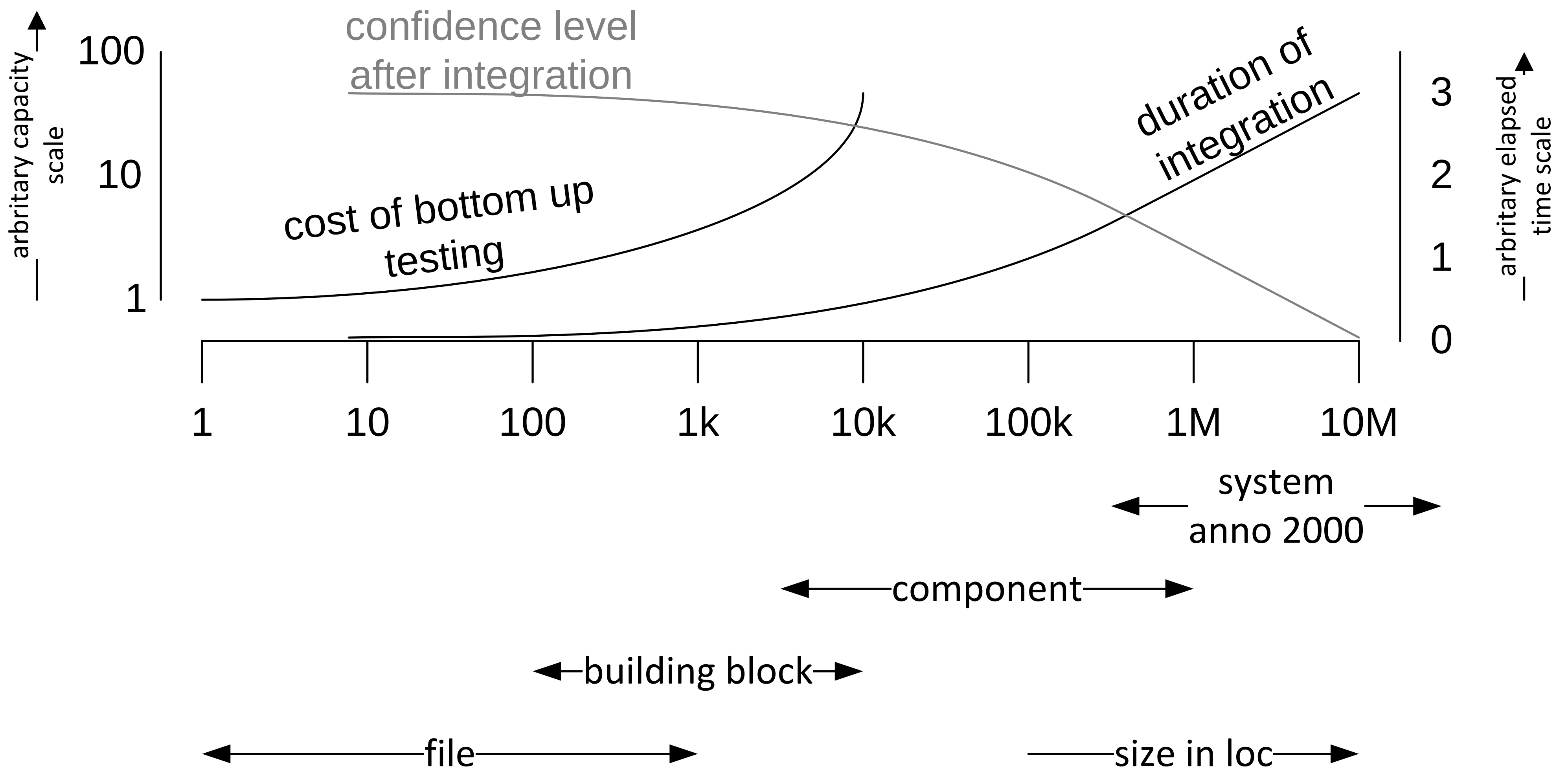
Nr Components vs Nr of Architects; Less Naive

		Capacity of architects c	10	20	40
Number of components n	Number of relations $r = n\sqrt{n}$	weight w	Number of Architects $a = (r * w)/c$		
2	3	12	3	2	1
4	8	9	7	4	2
10	32	4	14	7	3
20	89	2	22	11	5
40	253	2	39	19	10
100	1000	1	114	57	28
300	5196	1	534	267	133
1000	31623	1	3176	1588	794

Field Deployment viewpoint

- granularity of sellable features and services
- lifecycle support
- internal logistics and production process

Integration and Test viewpoint


confidence level after integration
cost of bottom up testing
duration of integration
arbritary capacity scale
arbritary elapsed time scale
1k
10k
100k
1M
10M
system anno 2000
component
building block
file
size in loc