



Estimation





Why estimate?

“A good estimate is an estimate that provides a clear enough view of the project reality to allow the project leadership to make good decisions about how to control the project to hit its targets.”

Software Estimation Steve McConnell



Estimates are not Forecasts

- Predictions are unreliable.
- Estimates are not promises.
- Estimates are not forecasts.
- Most teams do not do the same thing over and over again. Most new work is new, it requires knowledge discovery. Estimating things you don't know is not really possible.
- Treat Estimates as a rough guide only.



Estimation Approaches

- Historical
 - Regression from Past Data
- Counting:
 - Function Points
 - RICEWF
 - Use Case Points
 - Scrum/Agile Objective Estimation
- Expert Judgement
 - Scrum/Agile Story Points



Issues

- Many approaches to estimation, each takes a different perspective;
- Often tied to a process, e.g. RUP, AGILE, SCRUM, SDM, Enterprise Arch;
- Tied to organisation level: e.g. Program to Project;
- Can estimate *the Problem Space* **But** the Impacts of environment and technology, *the Solution Space*, affect how efficiently that can be delivered.
- Error reduces as we get more information and project progresses.
- Measurement feedback is required to calibrate (reduce error) and determine how well an approach works.
- All require decomposition – but what kind ?



RICEWFS





A relative sizing 'currency' based on complexity counts of:

Element	Definition
R eports	Any methods to present data to user.
I nterfaces	System to system interactions.
C onfigurations	Setting up and changing profile of re-built application features. Creating a customised object.
E xtensions	Creating custom code.
W orkflow	Sequence of activity steps, logic to complete a business transaction.
F orms	Any methods to capture data from user, GUI, web form.



Example Epic Story

As a:

Customer

I want to:

Create an Order on the Portal for a new Service that is submitted to the Fulfilment System for delivery

So that:

My business has access to the Service



Story Component Estimation – Part 1

Create an Portal Order

Report Order
submission result

	R	I	C	E	W	F
Simple						
Medium	1					
Complex				1		1
Extreme						

Interface to the Legacy
Ordering System via
Service on ESB

	R	I	C	E	W	F
Simple						
Medium		1				
Complex						
Extreme						

These Estimates relate to
Portal



Story Component Estimation – Part 2

Create an ESB
Business Service and
Access Service Adapter

	R	I	C	E	W	F
Simple						
Medium		2	2			
Complex						
Extreme						

Expose Ordering API on
Legacy System

	R	I	C	E	W	F
Simple						
Medium		1				
Complex				1		
Extreme						

These estimates relate to
Infrastructure and other
systems



Complexity Rules

	Reports	Interfaces	Configurations	Extensions	Workflow	Forms
Simple	<ul style="list-style-type: none"> • Uses existing Dimensions and Facts from source systems • No more than 3 source tables required to create view • Source system has been used in other similar reports • Standard functions and sums used in calculations • Standard report breaks, totals, and charts/graphs. • Standard formatting of reports or dashboards • Source system has been used in other similar reports 	<ul style="list-style-type: none"> • 1 source system • Less than 5 data elements sourced from a single source system table • Simple translations to data elements (no conditional logic) • Prior integrations exist between source and destination systems (proven integration pattern) • Low business impact if interface fails • Minimal to zero downstream dependencies (e.g. batch interfaces with large execution time windows) 	<ul style="list-style-type: none"> • Single form/table accessed • Minimal business process logic required • No new configuration objects required • i.e. new List of values, etc. 	<ul style="list-style-type: none"> • Extension to systems behaviour by writing code in a customised object, which is an add-on to a vendor product. • Minimal business process logic required • No new customised objects required • i.e. add 2-3 lines of code to add field values into a screen 	<ul style="list-style-type: none"> • Interacts with a single business process • Less than 4 workflow steps and no more than two branches • Individual steps do not require coding • No significant form UI development required 	<ul style="list-style-type: none"> • Modify existing forms by adding or deleting fields • Create a single entry form with existing field types • 1-3 existing data source tables, use existing sql statements • Use existing lookup tables, views or lists for popup fields • Basic error correction (required field, length, number, etc.)
Medium	<ul style="list-style-type: none"> • Uses 50% existing Dimensions and Facts from similar reports from source system • Limited to 3 -5 source tables required to create view • Max 2-3 functions and sums used in calculations that are non standard • Max of 3-5 report breaks, totals, and charts/graphs per report • Some non -standard formatting but general structure the source system has used in other similar reports 	<ul style="list-style-type: none"> • 1-2 source systems • 5-10 data elements sourced from a single source system table • Less than 5 data element if sourced from multiple source system tables. • Some conditional logic although rules limited to 3 conditions per logic statement • System is more than 6 months in production. • Maximum of one downstream dependency 	<ul style="list-style-type: none"> • Modification of existing configuration objects • Some business process logic but limited to only two processes • Multiple form/table accessed but limited to 3 tables 	<ul style="list-style-type: none"> • Modification of existing customised objects, which are add-on to a vendor product. • Some business process logic but limited to only two processes • Multiple form/table accessed but limited to 3 tables 	<ul style="list-style-type: none"> • Could be instigated by multiple business processes but limited to 4 L6 process • 4-8 steps and only 2 – 4 branches per work flow. • Individual steps may require coding but this is less than 50% of the total number steps 	<ul style="list-style-type: none"> • Replace existing single form with added functionality • Create complex single form with new field types • Use 4-5 existing tables or 1 external interface, date time or currency conversions for retrieval or storage, create 1-2 sql statements • Derived or calculated lookup fields, modify existing lookup table or view • Complex single form error checking (cross-reference multiple fields, calculation, regular expressions, etc.) • Interactive updates (prior form submission) to one or two UI elements based on separated business logic
Complex	<ul style="list-style-type: none"> • New Dimensions and Facts from source systems with additional dimensions • Over 5 source tables required to create view. • Over 4 functions and sums used in calculations that are non standard and complex • Over 5 report breaks, totals, and charts/graphs per report • Non -standard formatting and customized reports 	<ul style="list-style-type: none"> • More that 2 source systems • Over 10 data elements sourced from a single source system table or from multiple source systems • Conditional logic with multiple logic statements, conclusions and conditions • System is less than 6 months in production. • Multiple downstream dependency 	<ul style="list-style-type: none"> • configuration of new objects • Complex business logic or standard business logic greater than 2. • Over 3 form/tables accessed 	<ul style="list-style-type: none"> • customisation of new objects, which are add-on to a vendor product. • Complex business logic or standard business logic greater than 2. • Over 3 form/tables accessed 	<ul style="list-style-type: none"> • Instigated by multiple business processes more than 4 L6 process • More than 8 steps per workflow with multiple branches > 4. • Individual steps require significant coding 	<ul style="list-style-type: none"> • Streamline multiple forms into single form or split existing large form • Create multiple forms and maintain user state between them • Use > 5 existing database tables, 2 or more external interfaces, create multiple sql statements • External sources for lookup fields, create lookup tables or views • Complex error checking (cross-reference multiple forms, complex business rule calculation, multiple external systems for validation) • Interactive updates (prior form submission) to > 2 UI elements based on separated business logic
Extreme	<ul style="list-style-type: none"> • New Dimensions and Facts from source systems with additional dimensions • Over 10 source tables required to create view. • Over 10 functions and sums used in calculations that are non standard and complex • Over 10 report breaks, totals, and charts/graphs per report • Non -standard formatting and customized reports 	<ul style="list-style-type: none"> • More than 5 source systems • Over 20 data elements sourced from a single source system table or from multiple source systems • Conditional logic with multiple logic statements, conclusions and conditions • System is news in production. • Multiple downstream dependency 	<ul style="list-style-type: none"> • configuration of greater than 10 new objects • Complex business logic or standard business logic greater than 10. • Over 10 form/tables accessed 	<ul style="list-style-type: none"> • customisation of objects, which are core of a vendor product. • More than 20 steps per workflow with multiple branches > 10. • Over 10 form/tables accessed 	<ul style="list-style-type: none"> • Instigated by multiple business processes more than 10 L6 process • More than 20 steps per workflow with multiple branches > 10. • Individual steps require significant coding 	<ul style="list-style-type: none"> • Streamline > 5 multiple forms into single form or split existing large form • Create > 5 multiple forms and maintain user state between them • Use > 10 existing database tables, 5 or more external interfaces, create multiple sql statements • External sources for lookup fields, create lookup tables or views • Complex error checking (cross-reference multiple forms, complex business rule calculation, multiple external systems for validation) • Interactive updates (prior form submission) to > 10 UI elements based on separated business logic



Ideal Man Days (Size to Cost)

RICEWF counts are translated into estimated ideal man days using a standard formula.

Provides a uniform measure that gives an indicative size of a collection of work items to fulfil an Epic User Story.

Weighting factors		2	10	1	5	5	2
		R	I	C	E	W	F
1	Simple	2	10	1	5	5	2
4	Medium	8	40	4	20	20	8
10	Complex	20	100	10	50	50	20
25	Extreme	50	250	25	125	125	50

RICEWF Score:	xxx	RICEWF error +/-:	x.x
------------------	-----	----------------------	-----



Costs Across Enterprise - Example

Work Package	Platform	RICEWF Costs
Create an Portal Order	Portal	70
Report Order submission result	Portal	8
Interface to the Legacy Ordering System via ESB Service	Portal	40
Create an ESB Business Service and Access Service Adapter	ESB Infrastructure	88
Expose Ordering on Legacy System	Fulfilment Backend	90



- Applied at Enterprise and Program Level to allow consideration and comparison of many disparate work packages across projects.
- Provides developer estimates (cost), excludes effort for:
 - Architecture, BAs , PMs etc.
 - Impacts of Environment and Technology
- Other approaches required at next level of detail.



Issues

- Experience is showing the man day estimates to be high for Portal.
 - E.g. Interface weights are high.
 - everything has to be ‘Simple’
- Still open to interpretation, e.g. do portal forms have ‘workflow’ ?
- Resolution: many portal items require even ‘simpler’ changes to an existing form, not a “1” or a “0”; need a 1/3.
- Is an extension, e.g. to manage 6 order states, 1 Complex or 6 Simple Extensions.
- Weightings need to be re-calibrated based on feedback.



Story Points





Scrum/Agile Story Points

- Do not estimate in terms of time.
- Use a more abstracted metric to quantify effort. E.g.
 - numeric sizing (1 through 10),
 - t-shirt sizes (XS, S, M, L, XL, XXL, XXXL),
 - the Fibonacci sequence (1, 2, 3, 5, 8, 13, 21, 34, etc.) These take into account the inaccuracy of large estimates.
- Planning Poker – to remove bias estimates are not revealed until a signal.



Issues

- Tendency to Average.
 - As Stories are same (Average) don't need to estimate just count items (N. Stories).
- Based on experience or in-experience.
- Subjective not objective.
- Requires generalists, experience with requirements, analysis, design, integration, architecture, coding, testing etc....
- May not be reproducible or comparable across teams.
- The 'why' is easily forgotten.



Use Case Points





Use Case Points (UCP)

Take a Use Case and consider:

- Actor Interaction Types
- Use Case Transaction Complexity

Consider impacts to Implementation:

- Environmental Factors (EF)
- Technical Complexity Factors (TCF)

*Estimate the Problem then weigh by
Environment and Technology Impacts on
Capability to Implement.*



Actor Type & Use Case Complexity

Interaction Type	Description	Value
Simple	Well Defined Interface	1
Average	Dynamic Interface	2
Complex	Human Interaction	3

Use Case	No. Of Transactions	No. Analysis Classes	Value
Simple	< 3	<5	5
Average	4 - 7	5 - 10	10
Complex	>=7 Rules	> 10	15



Technology Complexity Factors

Factor	Description	Weight
T1	Distributed System	2
T2	Response Adjectives	2
T3	End-user efficiency	1
T4	Complex Processing	1
T5	Reusable Code	1
T6	Easy to Install	0.5
T7	Easy to use	0.5
T8	Portable	2
T9	Easy to change	1
T10	Concurrent	1
T11	Security Features	1
T12	Access to Third Parties	1
T13	Special Training Required	1



Environmental Factors

Factor	Description	Weight
F1	Familiar with Development Process	1.5
F2	Application Experience	0.5
F3	Object-oriented Experience	1
F4	Lead Analyst Capability	0.5
F5	Motivation	1
F6	Stable Requirements	2
F7	Part-time Workers	-1
F8	Difficult Programming Language	2

A '0' means the factor is irrelevant to the project, a '5' rating means it is essential.



Factors and Formulas

- $UUCP = \text{Total Actor weight} + \text{Total Use Case weight}$
- $EF\text{-Factor} = \text{Sum of (Weight * Value) column}$
- $TFactor = \text{Sum of (Weight * Value)}$
- $EF = 1.4 + (-0.03 * EF\text{-Factor})$
- $TCF = 0.6 + (0.01 * TFactor)$
- $UCP = UUCP * TCF * ECF$



Scrum/Agile Objective Estimation





Scrum/Agile Objective Estimation

Take a Story and consider:

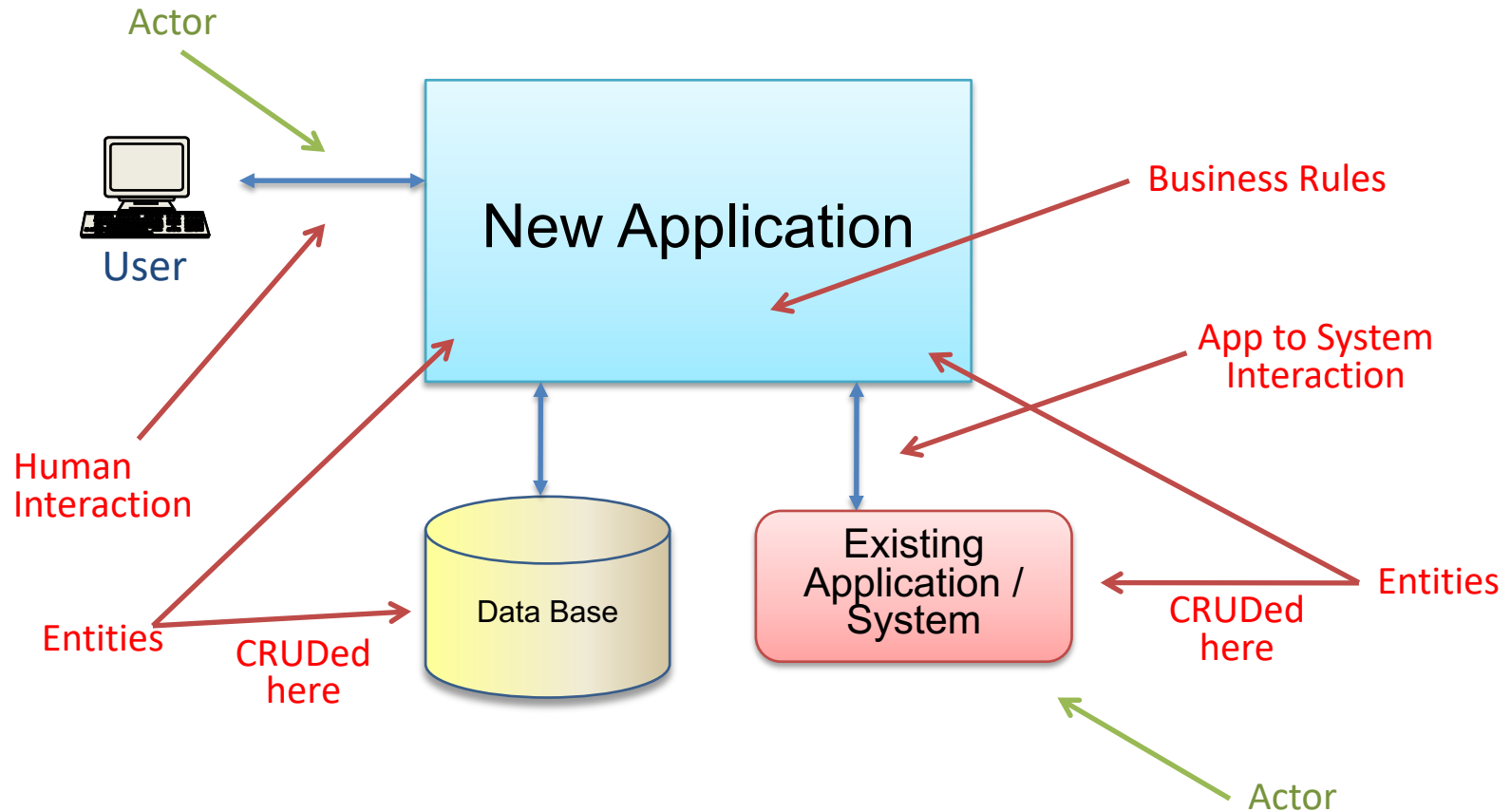
- Interaction Type
- Business Rules
- Number of Entities Manipulated
- CRUD

Problem Complexity = Total these to get
Unadjusted Points (UP).

Multiply by an Environmental Dimension (ED)
Factor to get Adjusted Points (AP) that takes
into account Capability to Implement.



Where Estimate Items come from:





Interaction Type & Business Rules

Interaction Type	Description	Value
Simple	Well Defined Interface	1
Average	Dynamic Interface	2
Complex	Human Interaction	3

Business Rules	Description	Value
Simple	1 Rule	1
Average	1-3 Rules	2
Complex	>3 Rules	3



Entities & CRUD

Entities	Description	Value
Simple	1 Entity	1
Average	1-3 Entities	2
Complex	>3 Entities	3

How we manipulate the Entities

CRUD Type	Description	Value
Simple	Read, Delete	1
Average	Create	2
Complex	Update	3



Environmental Dimension (ED)

Given we can estimate the problem how capable are we to implement it ?

Capability Dimensions:

- Organisational
- Development Infrastructure
- Team
- Technology
- Process
- Business



Dimension Questions (Value 0-2)

	Organisational Dimension
1	Have different departments worked successfully together on an Agile project previously?
2	Does some strong resistance exist within the organization with regards to Agile ?
3	Does great support for scrum exists between different departments within the company ?

	Dev. Infrastructure Dimension
1	Is automatic testing already in place and common practice ?
2	Is continuous integration testing already in place and a common practice ?
3	Is daily build environment already in place and a common practice ?



Dimension Questions (Value 0-2)

	Team Dimension
1	Is the team completely new to Scrum ?
2	Have the team member successfully worked together before ?
3	Do team members know well and appreciate one another ?

	Technology Dimension
1	Is the development team very experienced in the programming language ?
2	Are development team members very experienced in the technology to be employed ?
3	Is a scrum production environment already ready ?



Dimension Questions (Value 0-2)

	Process Dimension
1	Is Scrum the company adopted process framework ?
2	Is there a good support for scrum within the company ?
3	Is there strong resistance against scrum with the company ?

	Business Dimension
1	Is there a product owner fully available and completely engaged with the team ?
2	Is the product owner familiar with the scrum but has no practical experience ?
3	Has the product Owner successfully used scrum before ?



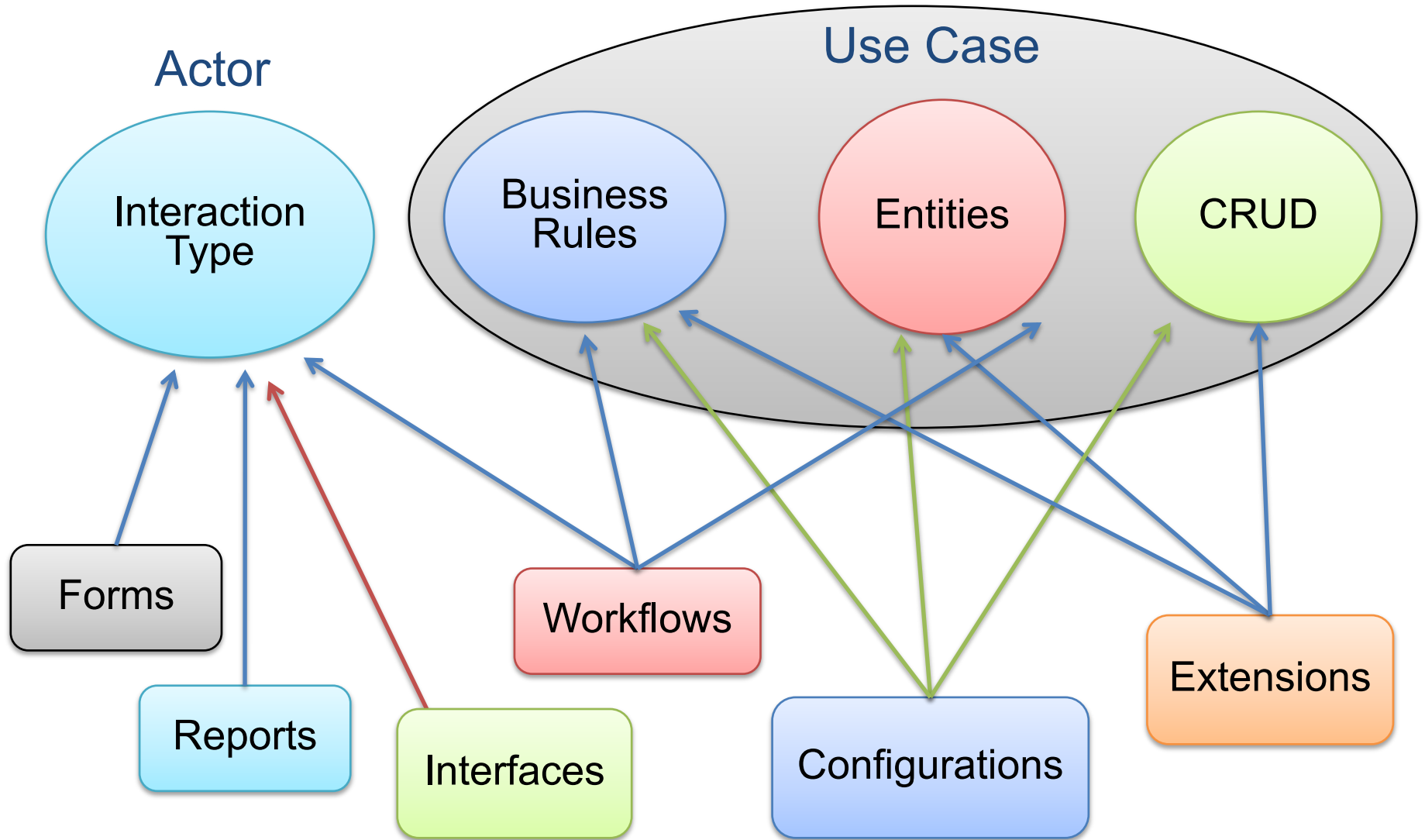
From ED to Coefficient

- ED Provides a value between 0 and 36
- Adjusted Points: $AP = UP * C$
- Where C is:

ED	Environment makes job	Value
0 - 11	Difficult	2
12 - 23	Neither difficult nor easy	1
24 - 36	Easy	1/2



Compare Use Case, Scrum & RICEWF





Summary

- Each approach decomposes the domain in a different way.
- RICEWFs used at higher program Epic level
 - Do not consider environment and technology impacts
 - Difficult to apply with Portal
- Use Scrum bases approaches at lower detailed Story levels.
 - Consider Objective approaches when possible