

Day 1
블루프리즘의 객체지향 RPA

What's

Robotic Process Automation



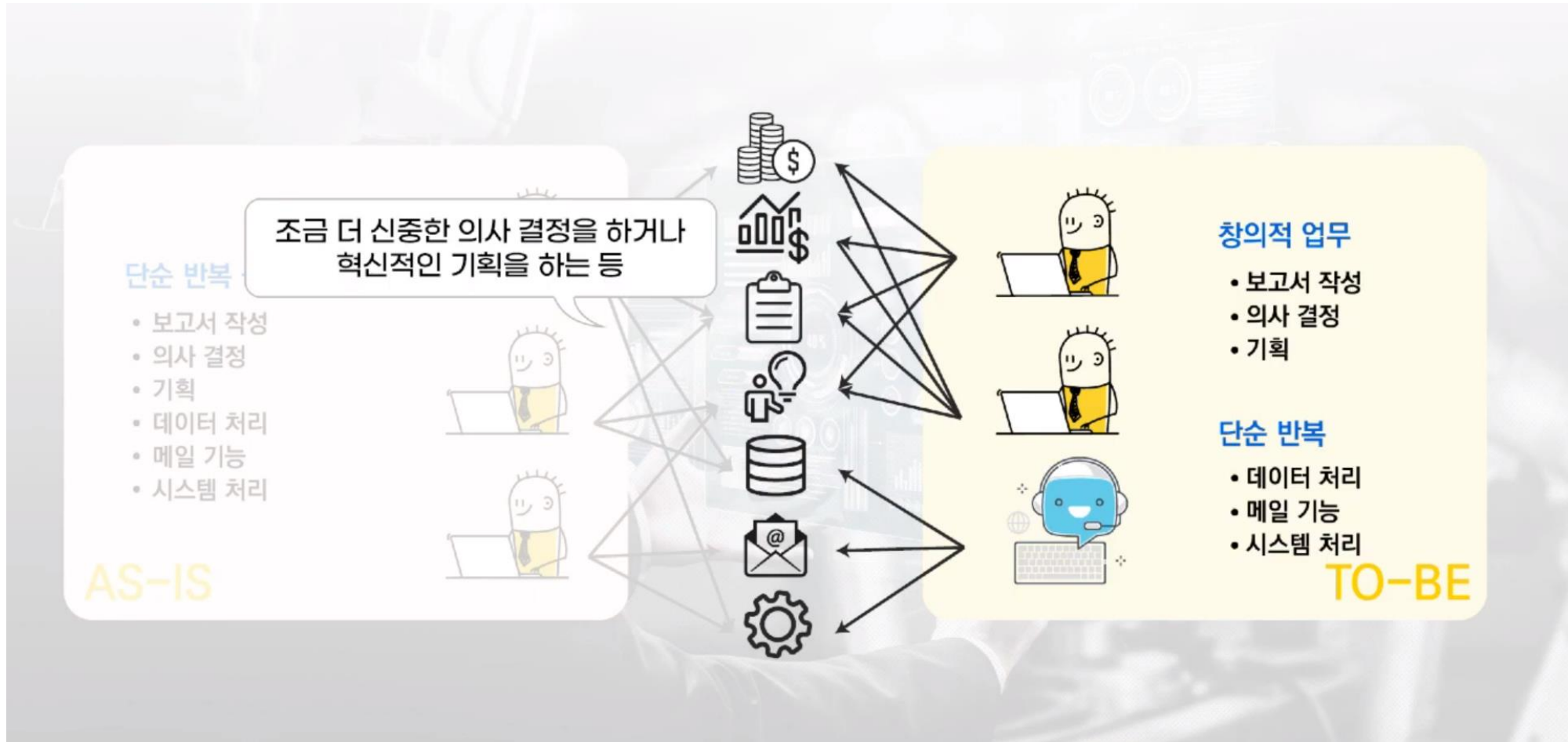
RPA, 업무 자동화란?

업무자동화란 로봇 프로세스 자동화 (Robotic Process Automation)이라 하고 이를 줄여 RPA라 부릅니다.

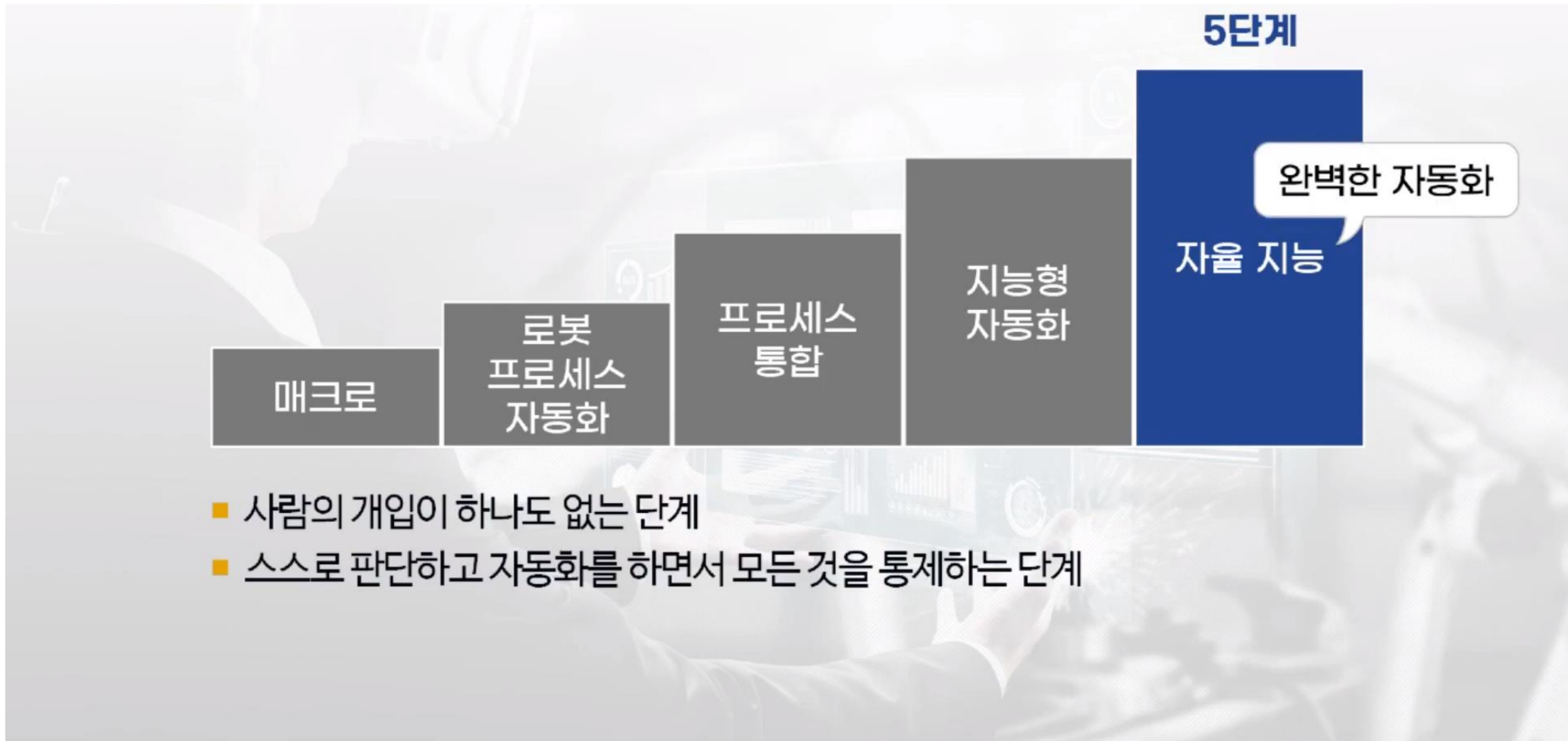
사람이 컴퓨터로 하는 단순 반복 작업을 자동으로 따라하는 소프트웨어로 만들어진 로봇을 말합니다. 공장에 있는 기계적인 로봇이 아니라 소프트웨어로 만든 프로그램을 말하는 것입니다. 가끔 로봇이라는 용어 때문에 기계를 떠올릴 수 있지만, 우리가 많이 접하는 엑셀, 아웃룩과 같은 소프트웨어 프로그램이라고 생각하시면 됩니다.

기존의 대부분의 업무들은 단순 반복 작업과 창의적 업무가 혼재 되어있는 상태입니다. 하지만, 일을 효율적으로 수행하려면 단순 반복적인 작업과 창의적인 업무는 분리되는 것이 좋습니다. 업무를 각각 분리해서 단순 반복적인 업무는 RPA를 통해서 수행하고 사람은 좀 더 창의적인 일에 집중할 수 있도록 업무 자동화가 도입되어야 합니다.

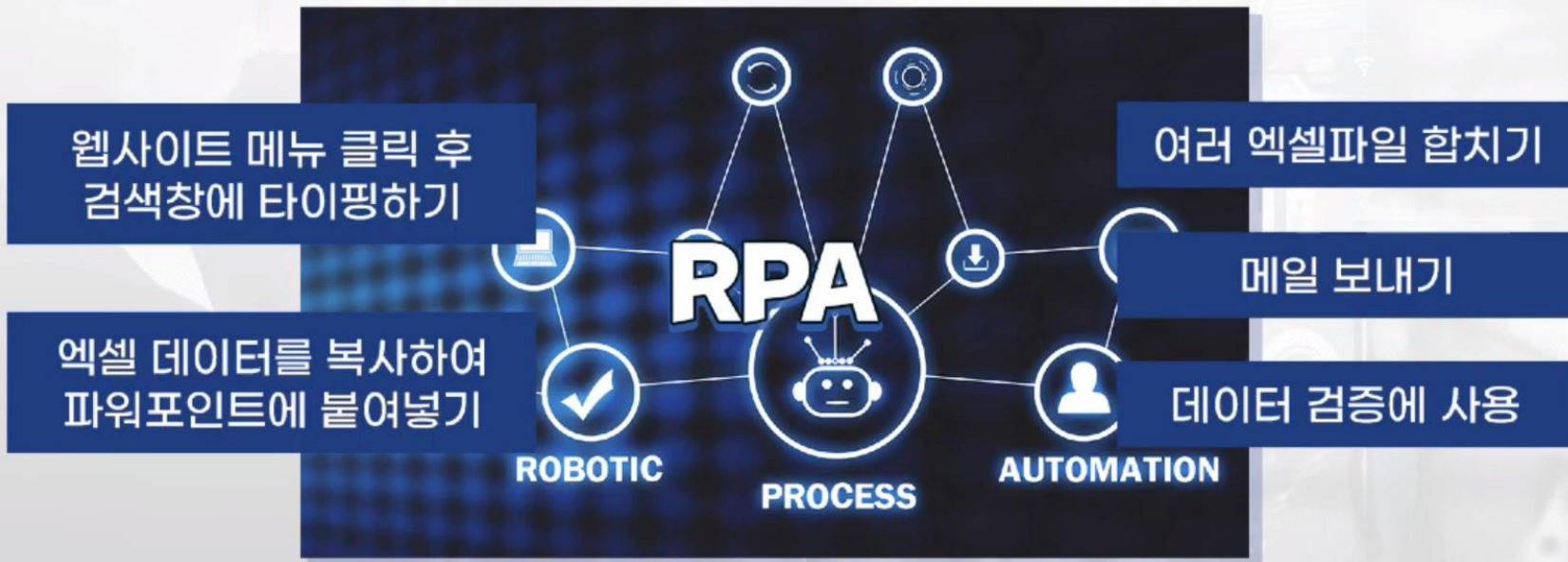
RPA가 필요한 이유



RPA의 진화 단계



업무 자동화란



**컴퓨터로 하는 업무 중 사람의 직관, 창의가 필요하지 않은
단순 반복 업무는 모두 자동화 가능**

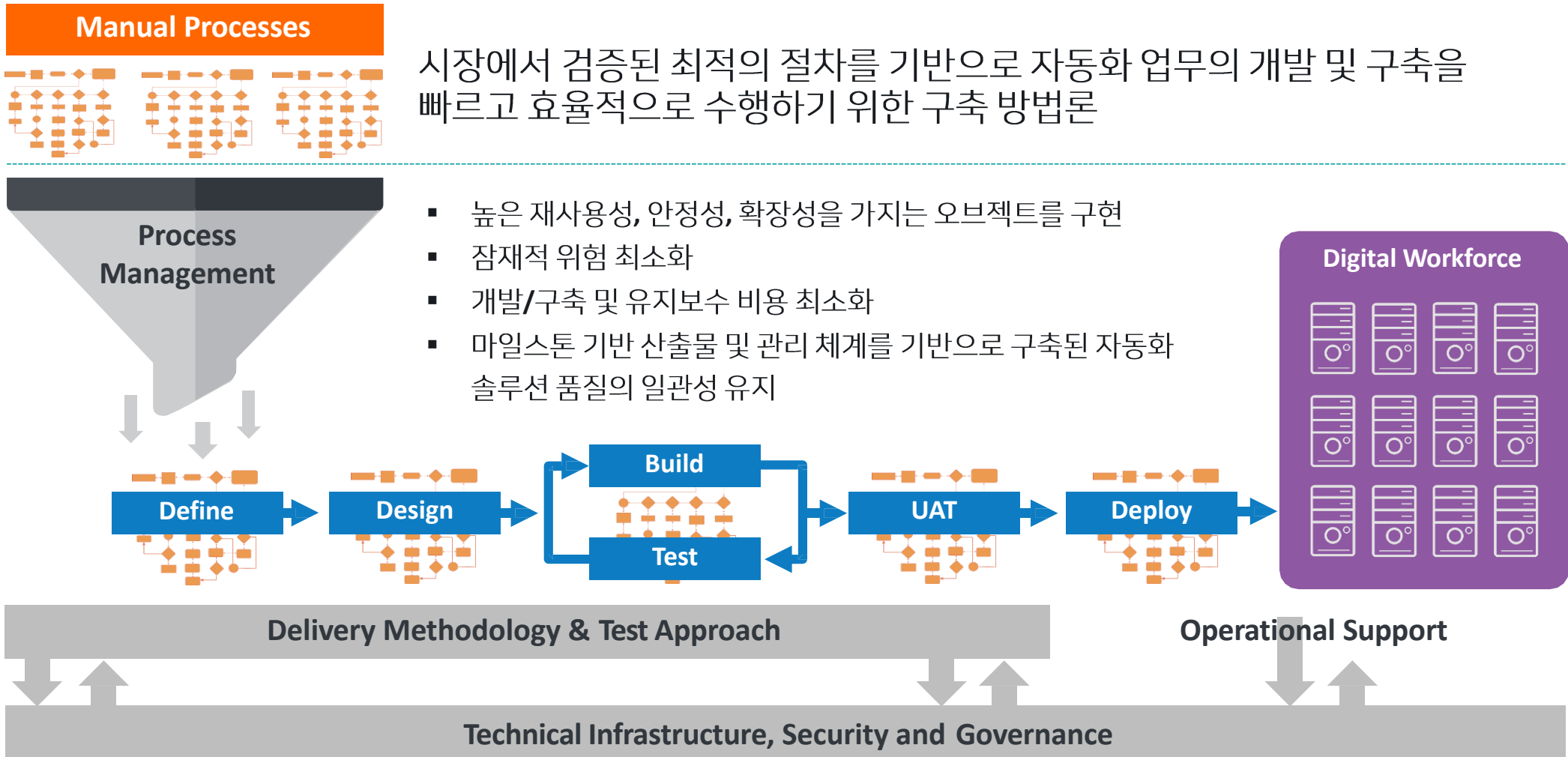
업무 자동화란

RPA 수행 가능 업무 예

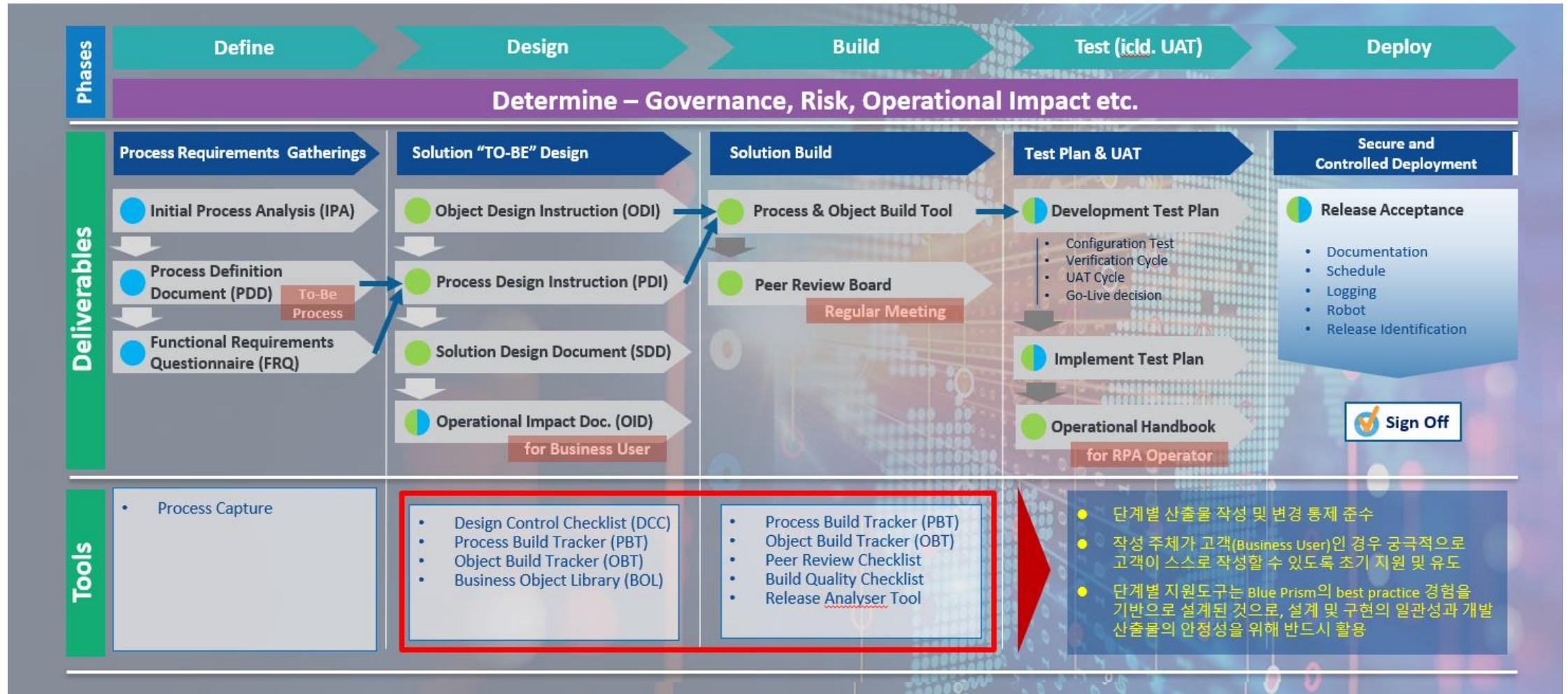
- 1 포털 사이트에서 뉴스를 찾아 뉴스레터를 만들어 사람들에게 이메일로 발송
- 2 쇼핑몰에 등록된 제품 정보를 수집해서 엑셀로 정리하고 통계내기
- 3 웹사이트에서 최신 정보를 검색해 파일을 다운로드하고 폴더를 만들어 정리
- 4 직원들에게 보내온 엑셀 파일을 하나의 파일로 합치고 통계 작성
- 5 수료증이나 영수증을 개별 PDF로 만들고 엑셀에 저장된 이메일 주소로 파일 보내기

사람이 컴퓨터로 하는 **단순업무는 모두 RPA가 대신 수행**

Blueprism Delivery Methodology - Lifecycle



Setting up the Delivery Methodology



Blueprism Delivery Methodology – Key Deliverables

M: Mandatory, O: Optional, R: Strongly Recommended

	STAGE	DELIVERABLES	DESCRIPTIONS	M/O	REMARKS
1.1	Define	Initial Process Analysis (IPA)	각 자동화 대상 업무별 절차, 자동화 범위, 복잡성, 예상 공수, 자동화 시 기대효과 등	M	
1.2		Application Assessment	고객의 응용서비스에 대한 정보 요약	O	
2.1		Process Definition Document(PDD)	자동화 업무의 흐름도 도시 및 자동화 설계를 위한 요구사항 정의 - IPA에 기술된 기초내용을 토대로 보다 구체화.	M	병합 가능
2.2		Functional Requirement Questionnaire(FRQ)	업무적 요구사항을 만족하는 Unattended 형태의 자동화 솔루션을 설계하기 위한 기능 요구사항 정의	M	
3.1	Design	Object Design Instruction (ODI)	단위 오브젝트에 대한 상세 설명: e.g., 입출력 파라미터, Actions, 등	M	
3.2		Process Design Instruction (PDI)	구현될 블루 프리즘 솔루션의 프로세스, 컴포넌트, 비즈니스 오브젝트, 워크 큐, 암호에 대해 상세 기술	M	
3.3		Solution Design Document (SDD)	솔루션, 프로세스, 오브젝트 구성 다이어그램, 운영 방법, 데이터 보안 상세 정의	M	
3.4		Operational Impact Document (OID)	자동화 솔루션 운용을 위한 관련 부서(e.g, IT, Operation) 요구사항	M	
3.5		Design Authority(DA)	개발 컴포넌트간 적합성을 보장하기 위해 관련 프로세스 및 오브젝트의 통합 저장소를 유지하고 각 컴포넌트별 변경을 통제	R	별도 톨 사용. 일관성 및 고품질 개발 산출물을 위해 매우 중요.
4.1	Build (Dev)	Peer Review Board	개발 산출물의 적정성에 대한 점검 톨로 개발팀 내 개발자 상호간 또는 각 개발자가 수행.	R	별도 톨 사용. 일관성 및 고품질 개발 산출물을 위해 매우 중요.
4.2		Build Review Checklist	프로세스의 안정성, 확장성, 유연성, 관리용이성을 향상시키기 위한 점검 톨	R	
4.4		Development Test Plans	검증 및 인수 시험 항목 및 절차 정의	M	병합 가능
5.1	Test	Verification Test Plan	개발확인 시험 계획서(시험 환경, 범위, 합격/실패 기준 등에 관한 제반 사항)	O	
5.2		UAT Plan	인수 시험 항목 및 절차 정의	O	
6.1	Deploy	Release Note	신규 릴리즈된 패키지에 대한 상세 설명	M	
6.2		Operational Handbook	운영 매뉴얼	M	
6.3		Process Release Acceptance	신규 릴리즈의 상용 적용을 위한 산출물 검토 및 승인	O	
7.1	Technical Infrastructure, Security, Governance	Archive and Backup Policy	로그 레벨 설정, 프로세스 및 오브젝트 백업 정책 및 가이드	R	병합 가능
7.2		Logging Configuration Policy	구동 환경 별 로그레벨 설정 정보 및 가이드	O	
7.3		Security Policy and Procedures	보안 정책 및 절차 정의, 각 직무 별 보안 및 권한 설정 정보	R	

Contents

01

객체지향 RPA

객체지향 프로그래밍(OOP)에 대한 이해를 바탕으로 블루프리즘이 객체지향 RPA를 어떻게 실현 하는지 알아봅니다.

02

오브젝트 & 프로세스

객체지향 RPA를 실현하기 위한 핵심 개념, '비즈니스 오브젝트'에 대해 알아보고, 좋은 프로세스란 무엇인지 그리고 이를 위해 블루프리즘은 어떤 설계방법 및 기능을 제공하는지 알아봅니다.

03

워크큐 & 프로세스 템플릿

기업의 자동화 프로세스를 표준화 하기 위해 블루프리즘이 제공하는 워크큐와 프로세스 템플릿을 어떻게 활용하는지에 대해 알아봅니다.

What's

Object-oriented Programming

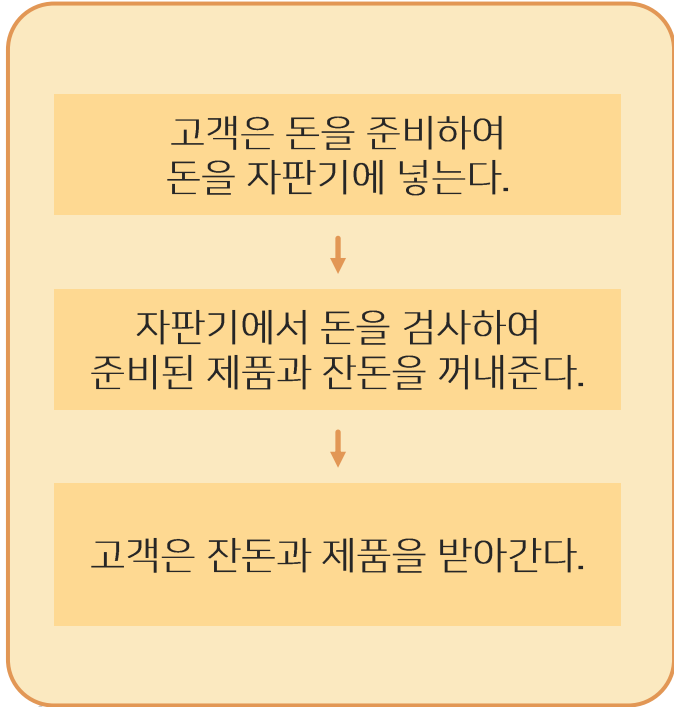


객체지향 프로그래밍이란?

컴퓨터 프로그래밍을 하기 위한 언어에는 많은 종류가 있다. 이러한 언어들은 크게 2가지로 나누어지는데, 객체 지향 언어와 절차 지향 언어이다. “객체 지향이란 실제 세계를 모델링하여 소프트웨어를 개발하는 방법으로서, 객체 지향 프로그래밍에서는 데이터와 절차를 하나의 덩어리로 묶어서 생각한다. 이는 마치 컴퓨터 부품을 하나씩 사다가 컴퓨터를 조립하는 것과 같은 방법이다.”

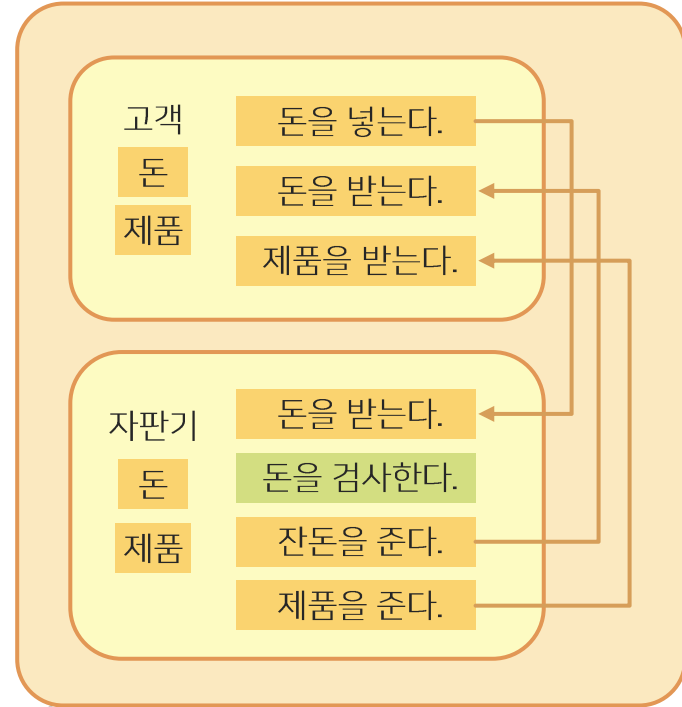
객체(object)란 보고 만질 수 있는 것, 지성적으로 이해할 수 있는 것, 생각이나 행동이 추구하는 바를 말한다. 또는 문제영역에서 잘 정의된 역할을 갖고 있는 각각에 대해서 구별할 수 있는 품목(item), 단위(unit), 개체(entity)라 정의하기도 하며 단순히, 정의된 경계를 갖고 구별되는 어떤 것이라 말할 수도 있다.

다시 말해서 객체는 학생, 교실, 책 같은 생각할 수 있는 모든 사물이나 공부, 수학 같은 개념상으로 존재하는 것 등 모든 것이 될 수 있다. 좀 더 구체적으로, 문제영역에 속한 사물 중에 관리의 필요성이 있거나 중요한 개념이라면 더 좋은 객체가 될 수 있다. 시스템의 관점에서 본다면 어떤 상태(state)를 나타내는 데이터의 구조와 동작을 수행하는 연산(operation)으로 이루어진 프로그램의 한 요소이다. 여기서의 연산을 객체 지향에선 메소드(method)라고 한다. 이렇게 객체의 상태는 데이터에 의해 결정되고 동작은 메소드에 의해 결정된다.



절차지향 분석

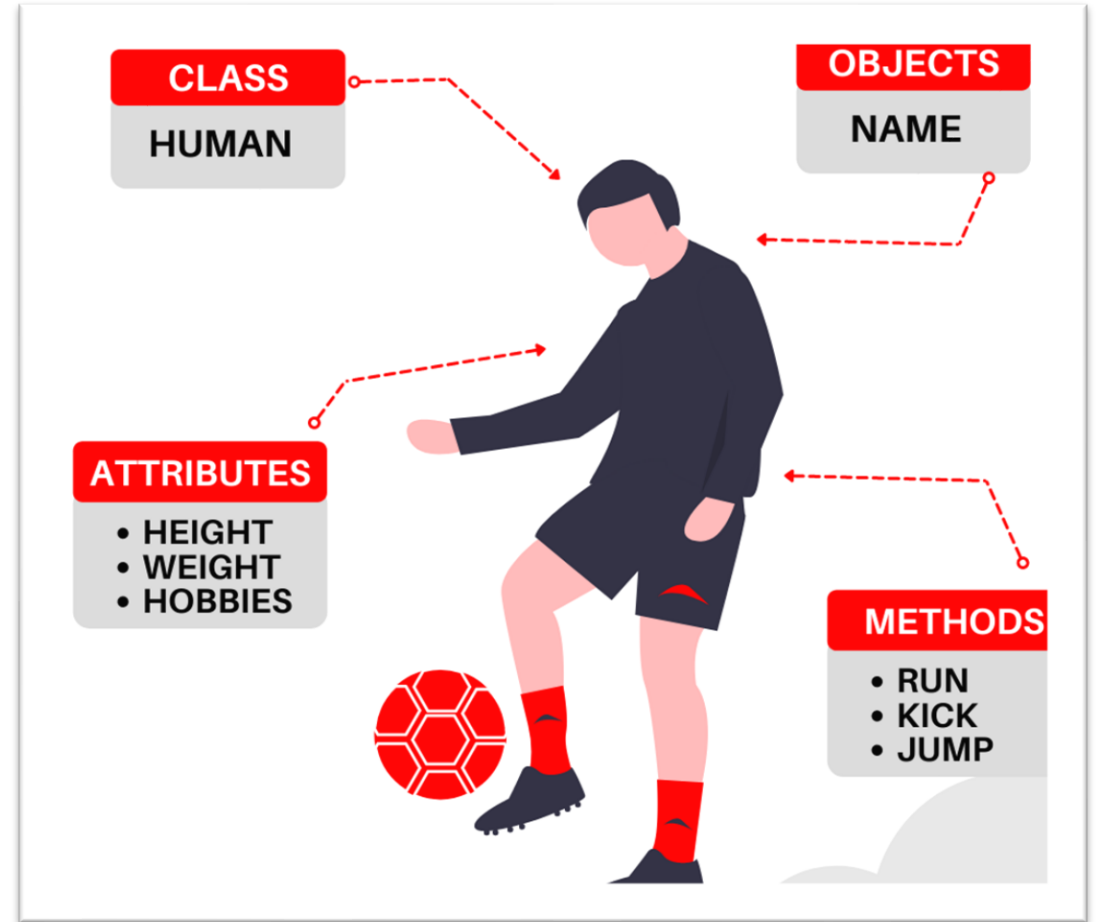
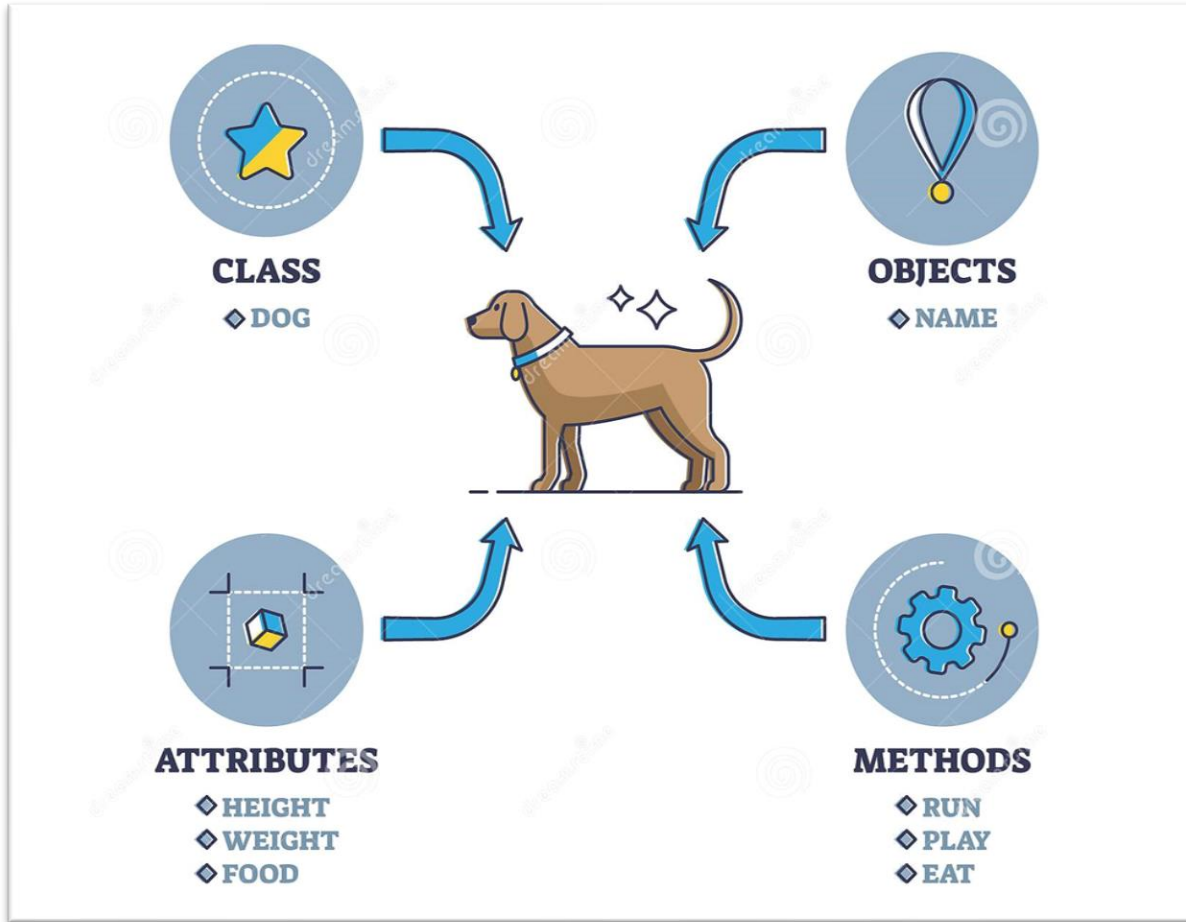
데이터를 중심으로 순차적 실행에 초점을 맞추고 코드를 작성한다. 돈을 지불하는 절차, 계산하는 절차, 제품을 받는 절차를 중점으로 함수가 구성되었다.



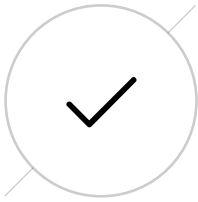
객체지향 분석

프로세스의 주요 행위자인 고객과 자판기를 객체로 설정하고 돈과 제품에 대한 관계에 따라 객체를 중심으로 함수를 작성한다.

객체(Object)의 예시

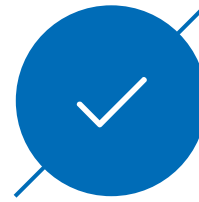


객체지향 프로그래밍의 이점



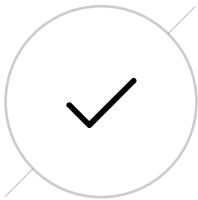
재사용성

상속을 통해서 프로그래밍 시 코드의 재사용을 높일 수 있습니다.



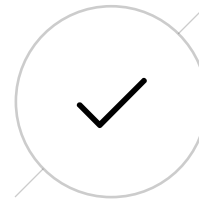
생산성 향상

잘 설계된 클래스를 만들어서 독립적인 객체를 사용함으로써 개발의 생산성을 향상시킵니다. ctrl + v 실행은 나중에 수정사항이 생겼을 때 일일이 찾아서 바꿔주어야 되지만, 클래스는 한 곳에서만 바꾸면 됩니다.



자연적인 모델링

실제 세상의 사물들을 하나의 객체로 생각하며 프로그래밍이 이루어지기 때문에 클래스를 구현할 때 일상생활의 모습과 대비시키며 자연스럽게 구현할 수 있습니다.



유지보수의 우수성

캡슐화로 인해 모듈끼리의 의존성이 떨어집니다. 그렇기에 프로그램 추가 및 수정과 같은 유지보수에 용이합니다.



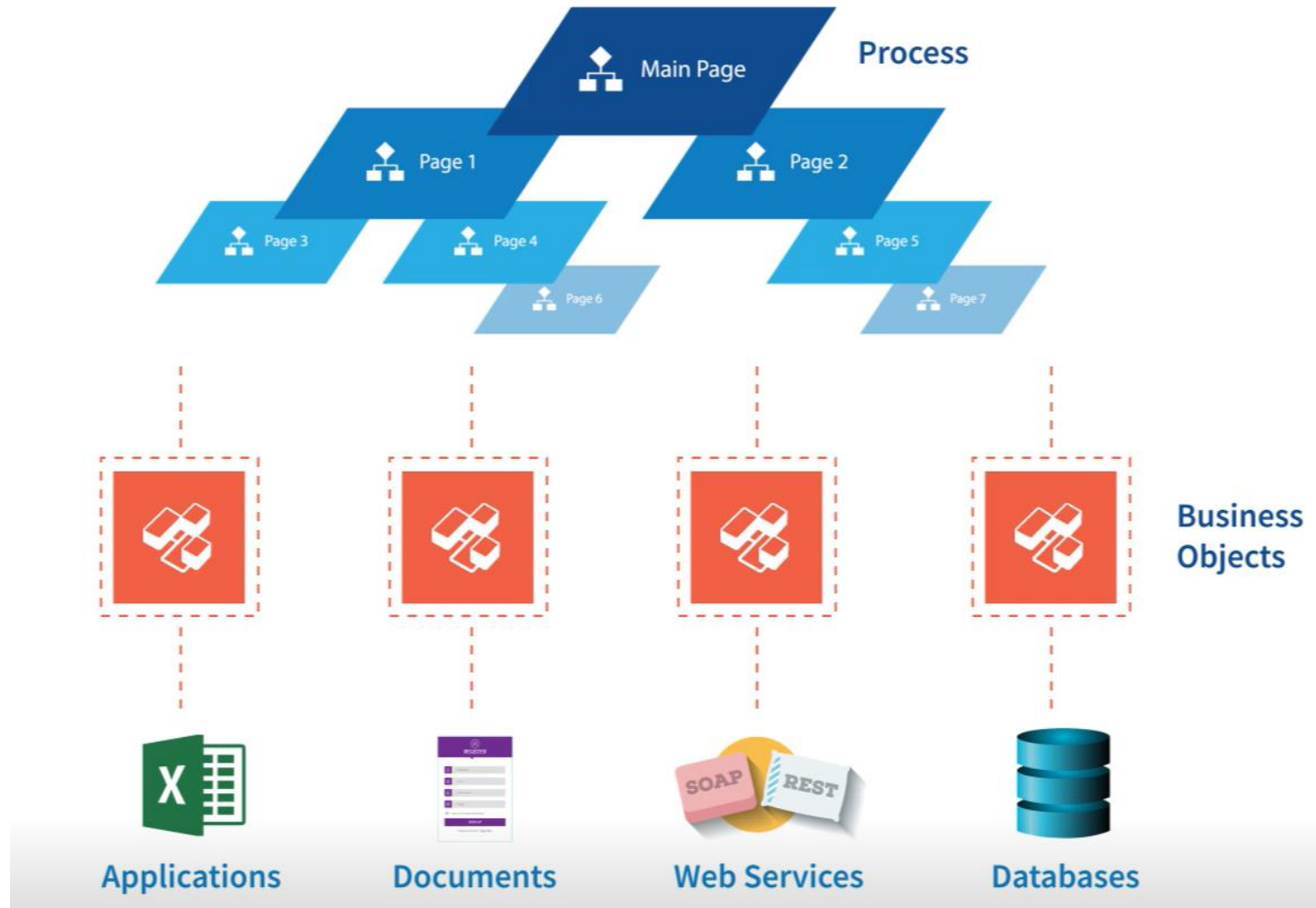
Blueprism

RPA

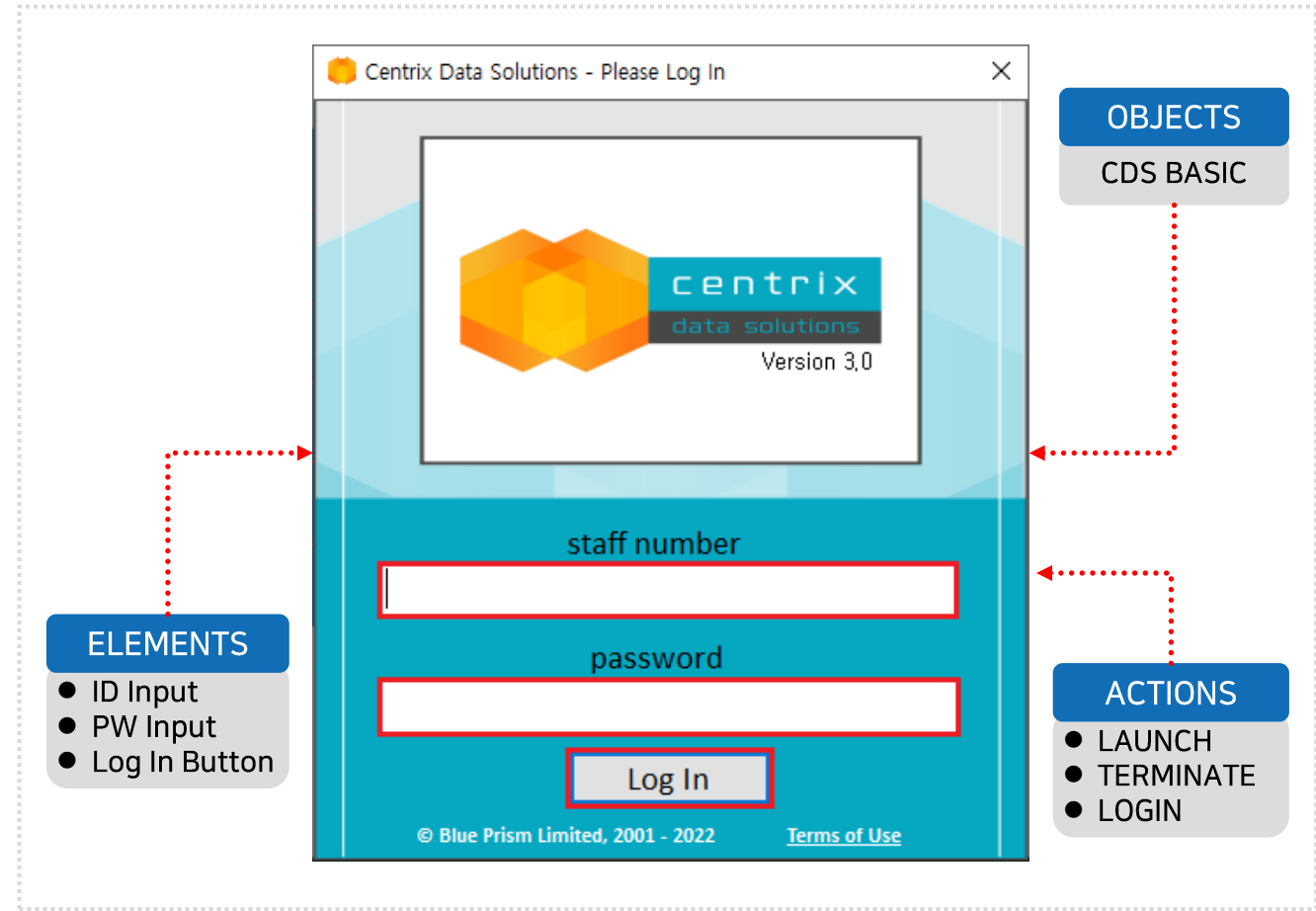
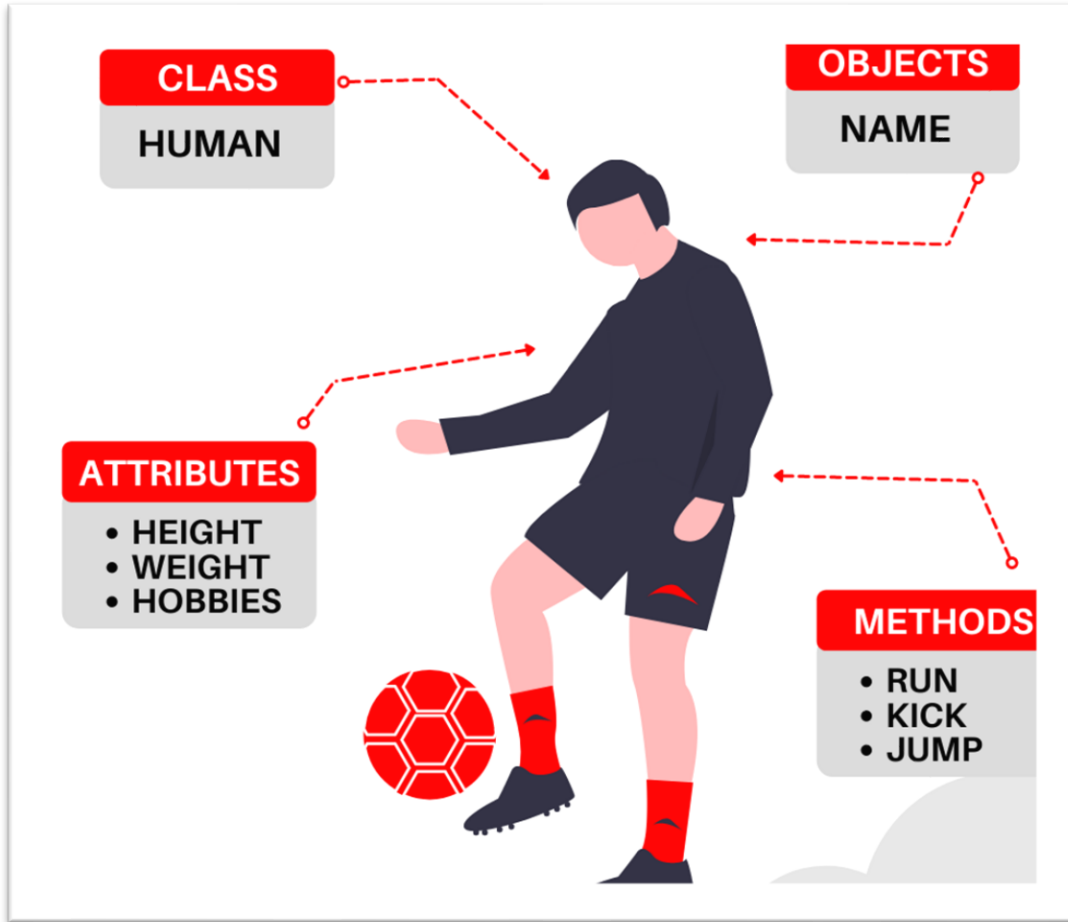
블루프리즘의 객체지향 RPA란?

블루프리즘은 객체지향 RPA를 위해 오브젝트와 프로세스를 분리하고, 오브젝트는 객체지향 프로그래밍의 방식을 따르도록 설계, 프로세스는 작업의 흐름만을 담당하도록 설계.

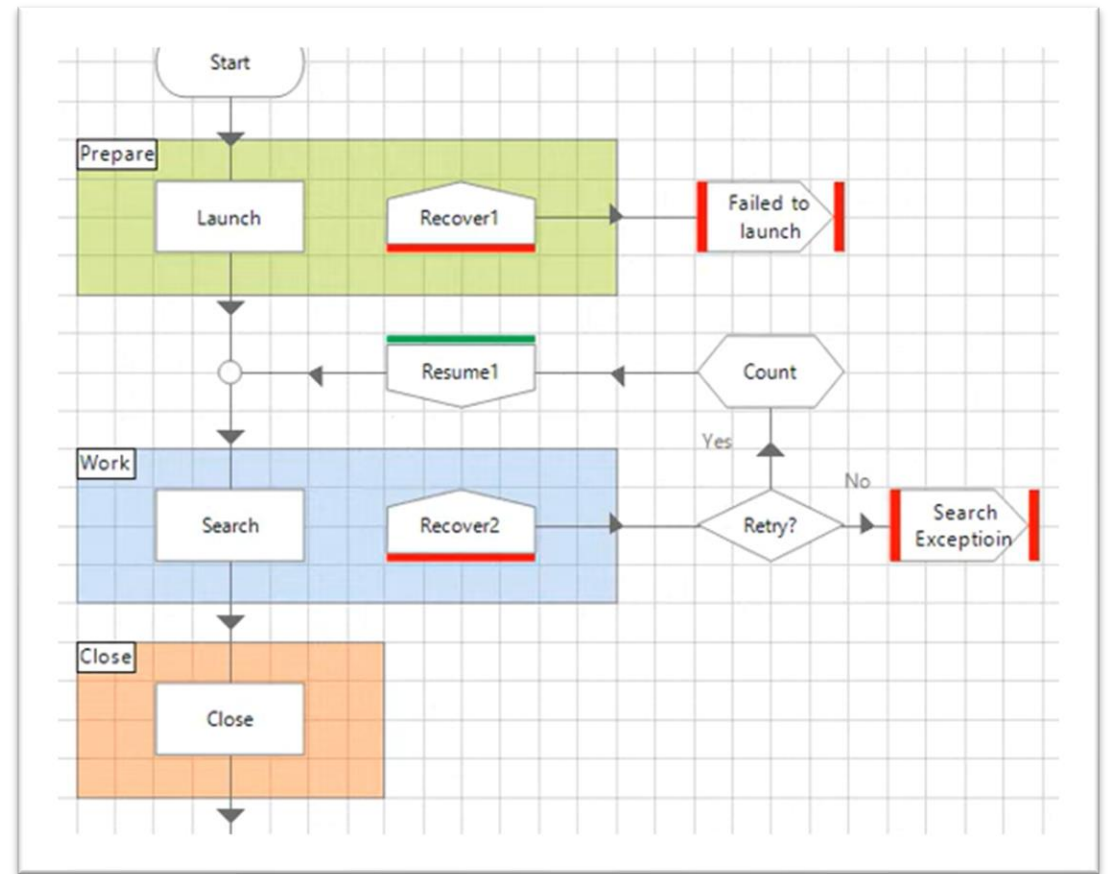
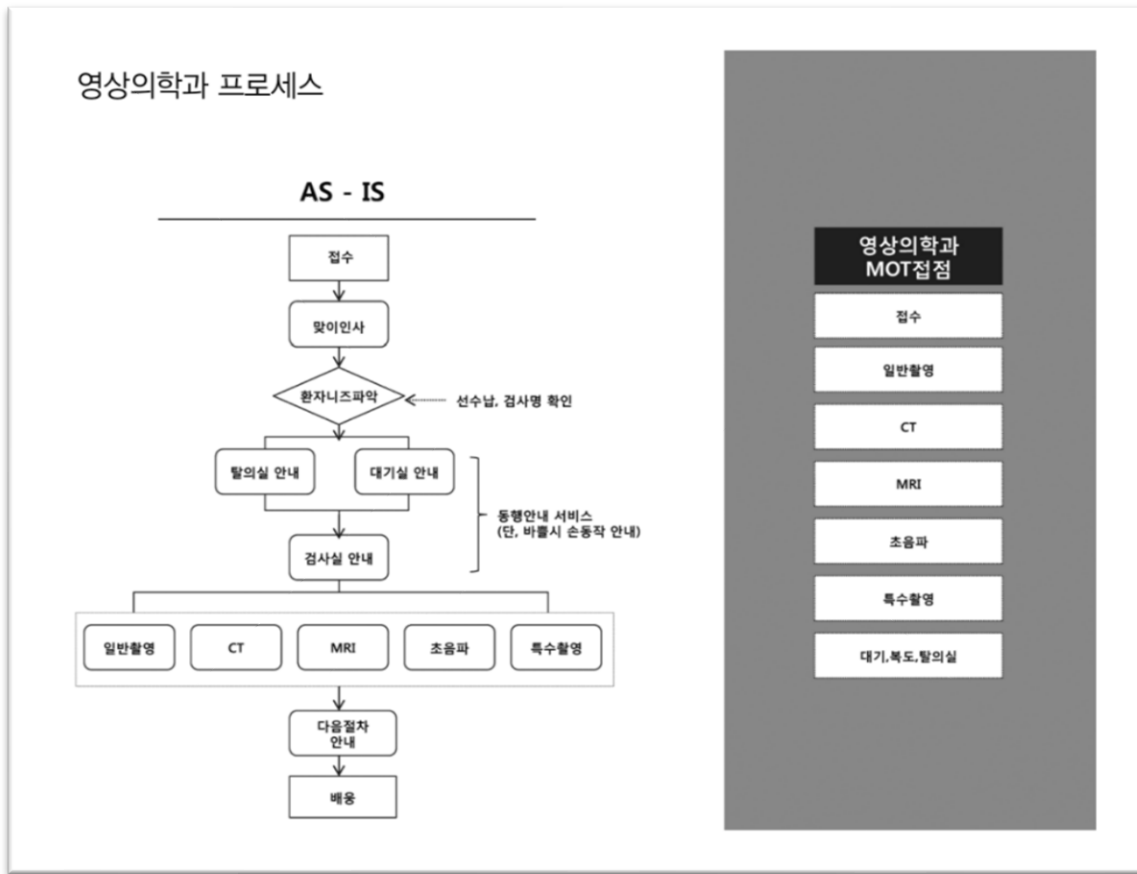
블루프리즘 프로세스의 구성



비즈니스 오브젝트

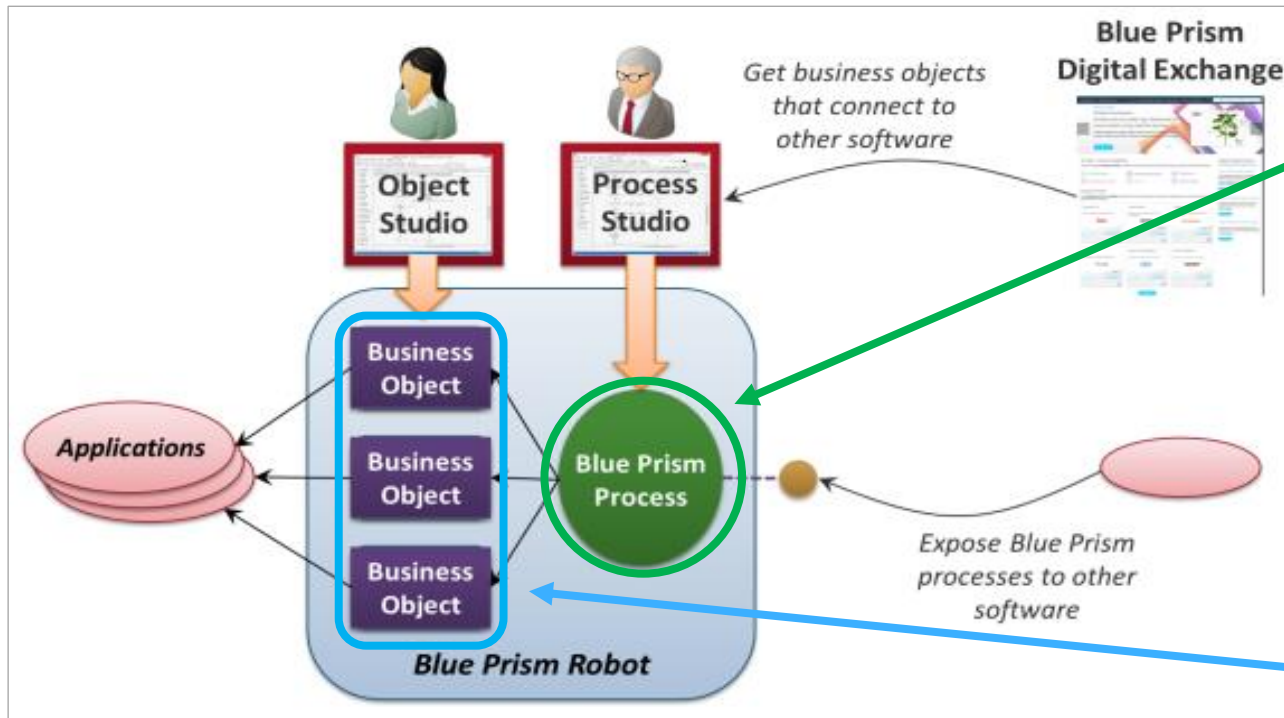


블루프리즘 프로세스

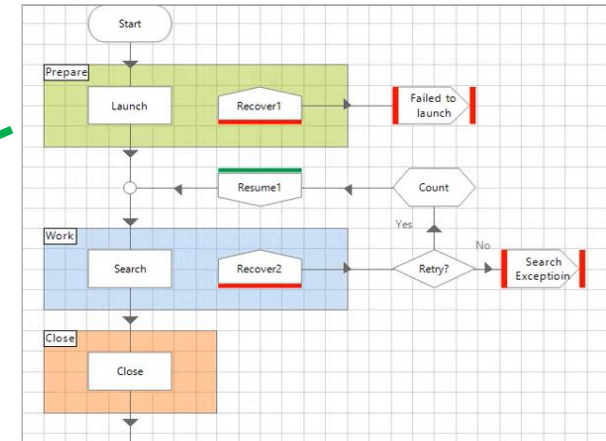


블루프리즘 프로세스의 구성

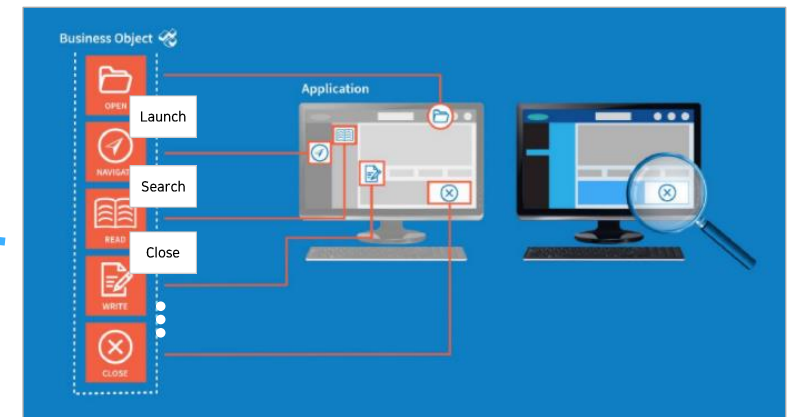
- Object-oriented RPA in Blue Prism



- Blue Prism Process



- Business Object



블루프리즘 프로세스의 구성



- Business Object



- Business Objects



- Business Process

What's

Blueprism Business Objects



비즈니스 오브젝트란?

비즈니스 오브젝트는 프로세스 흐름도에서 웹 서비스와 함께 외부 시스템과 상호 작용하는 수단을 제공한다.

비즈니스 오브젝트는 프로세스에서 호출할 수 있는 기능을 Action으로서 제공한다. 예를 들어, 사내 시스템에 로그인 하는 기능을 수행하는 Action이 있다면 프로세스에서는 로그인 하고자 하는 계정정보를 매개변수로 넘겨주며 해당 Action을 호출하면, 대상 시스템에 로그인 하는 기능을 수행할 수 있다.

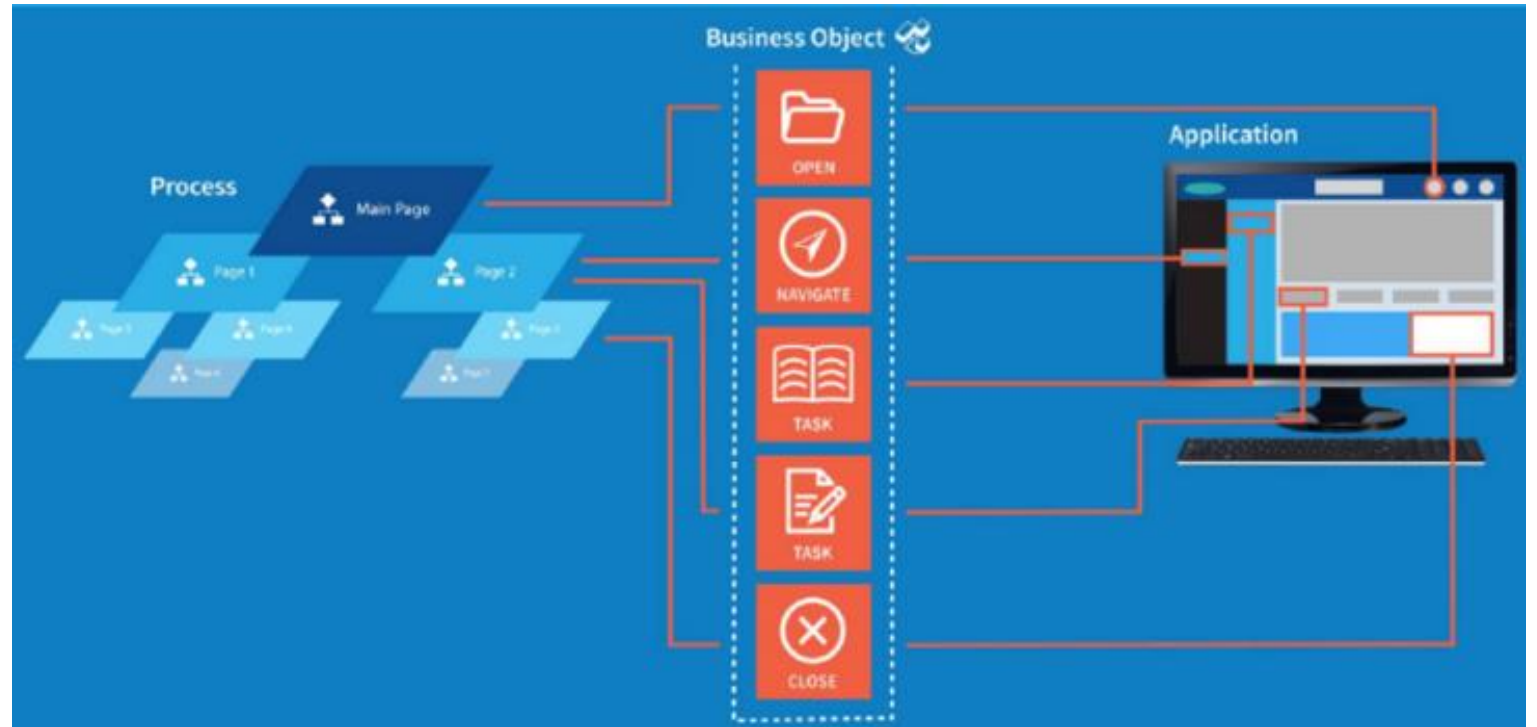
이러한 방식으로 비즈니스 오브젝트에 Action을 통해 기능을 제공하면 동일한 프로세스에서 여러 번 사용 되거나 다른 프로세스들 에서도 여러 번 사용 될 수 있다. 이는 객체지향 RPA의 기본 원리이며, 이를 통해 재사용성을 높일 수 있다.

블루프리즘의 비즈니스 오브젝트 구현은 프로세스 구현과 분리된다. 이는 대상 시스템을 새 시스템으로 교체하는 경우 비즈니스 오브젝트의 재구현이 필요하지만 프로세스의 변경은 필요 없음을 의미한다.

이러한 이점을 최대한 활용하려면 구현에 있어 비즈니스 로직을 분리해야 한다. 대상 시스템들과 상호작용 하기위해 비즈니스 오브젝트를 사용하고, 데이터 흐름 및 큐 아이템 처리 등을 위해 프로세스를 사용한다.

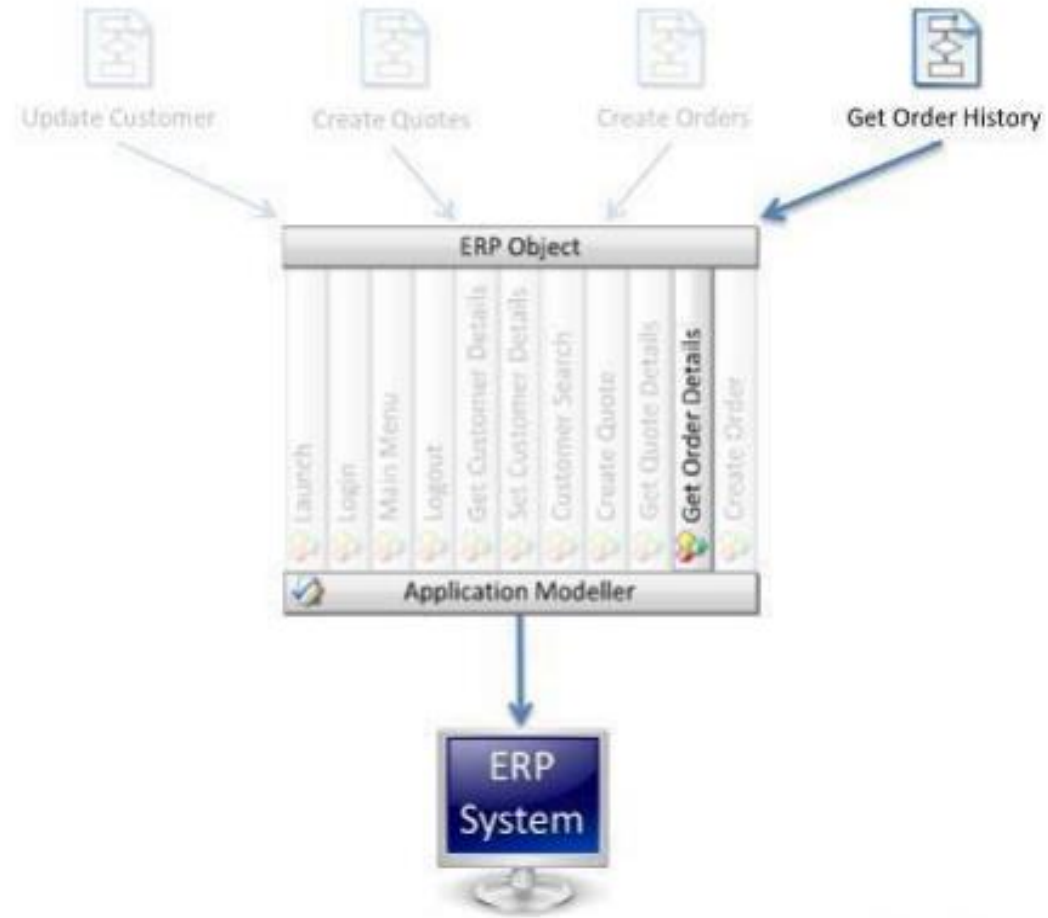
오브젝트의 특징

- 어플리케이션 모델러를 통해 대상 어플리케이션을 모델링
- 단독으로 사용되지 않고 프로세스의 일부로 사용 됨
- Action이라 불리는 계층이 없는 단순한 구조로 단일 기능만을 제공
- 각 Action은 대상 어플리케이션과 상호작용. 프로세스는 이를 호출하여 상호작용



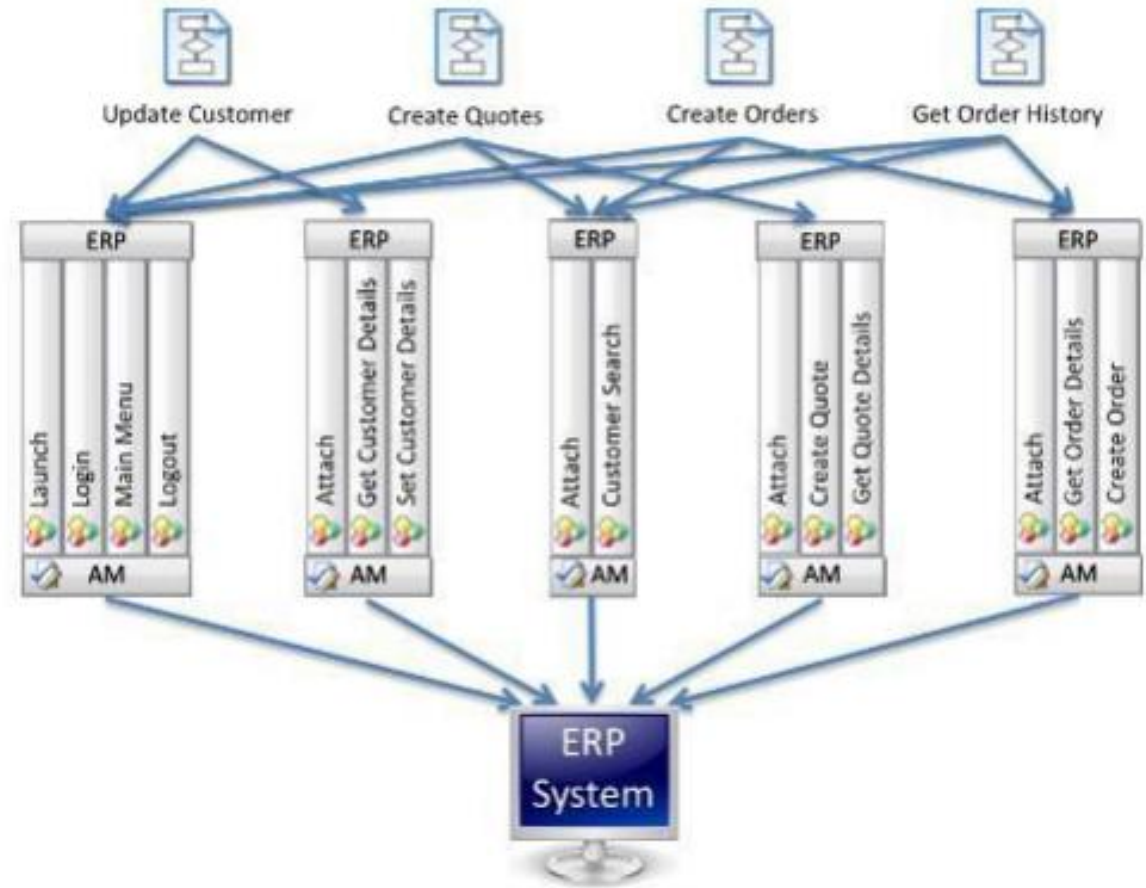
단일 오브젝트 디자인

- 한 번에 한 명의 개발자만 오브젝트를 작업할 수 있음
- 불필요한 작업으로 인한 메모리 낭비
- 오브젝트의 수정은 곧 모든 프로세스에 대한 변경
- 필요 이상으로 큰 오브젝트로 인한 DB 비효율성 증가
- ❖ 프로덕션으로 이관되지 않을, 테스트 및 소규모 PoC 등을 위한 프로젝트에 적합



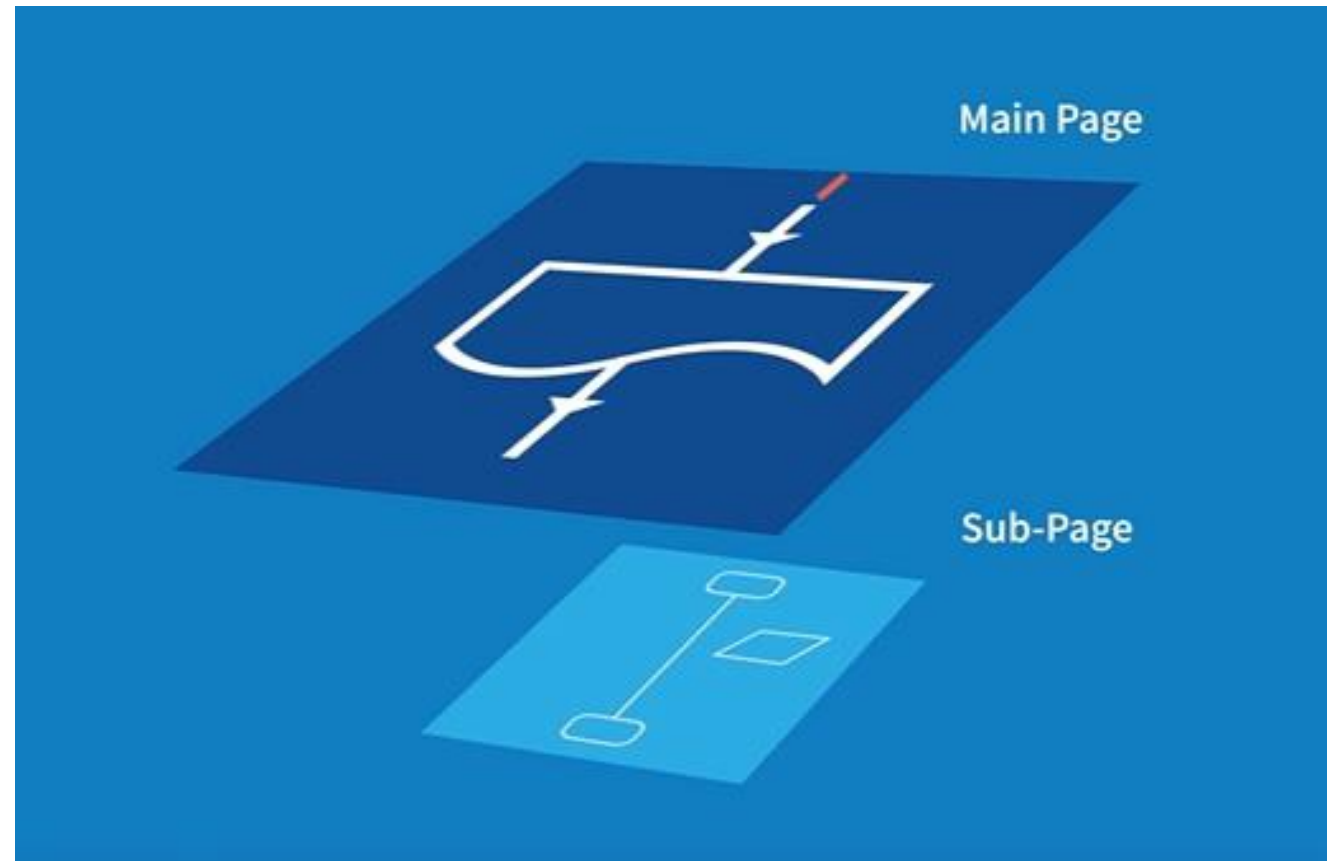
다중 오브젝트 디자인

- 한 번에 여러 명의 개발자가 ERP 오브젝트를 작업
- 메모리 효율성 증가
- 오브젝트의 수정은 적은 프로세스에 영향
- 효율적인 데이터베이스 사용
- ❖ 프로덕션에 배포 될 프로세스나 여러 명의 개발자가 참여하는 PoC에 적합.



입력 및 출력 매개변수

- 프로세스의 Action Stage와 비즈니스 오브젝트의 Action Page 간의 제어
 - 비즈니스 오브젝트는 가능한 일반적으로 구현되어야 함
 - 비즈니스 오브젝트는 어떠한 비즈니스 데이터도 저장해서는 안 됨
 - 비즈니스 오브젝트에서 필요한 비즈니스 데이터는 입력 매개변수를 통해 전달받고, 필요 시 출력 매개변수로 데이터를 전달
- ❖ 위와 같이 구현 함으로서 오브젝트의 재사용 성이 보장 됨.

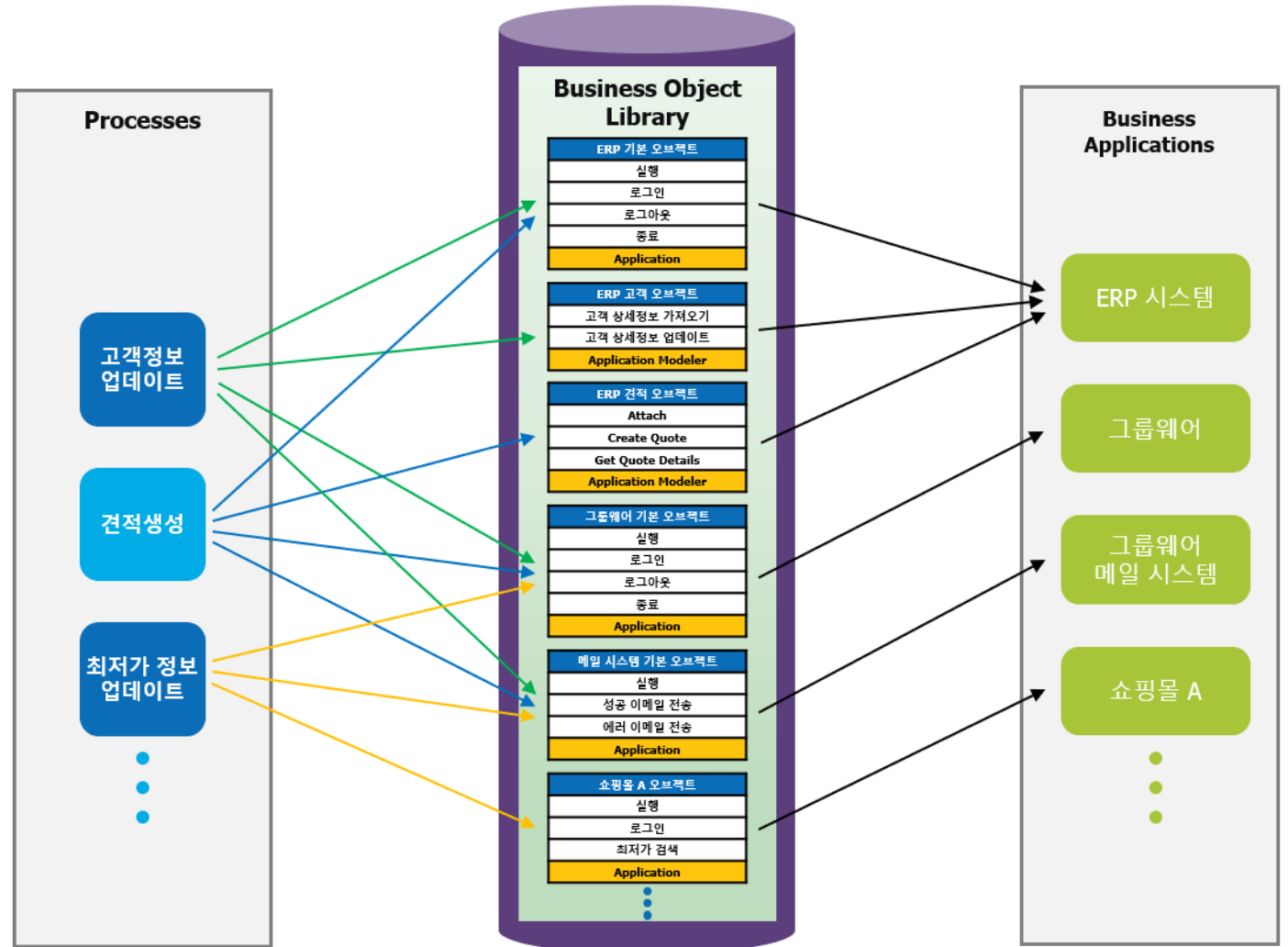


Business Object Library (BOL)

조직의 RPA가 확산됨에 따라 오브젝트는 많아지게 되고, 누적된 재사용성 높은 오브젝트들은 RPA의 확산 속도를 가속화 하기도 하지만, 많은 오브젝트를 잘 관리하는 것 또한 중요하다.

Business Object Library(BOL)는 기개발된 오브젝트들을 한 눈에 볼 수 있는 툴로, 설계 단계에서 기개발된 오브젝트를 재사용 할지, 추가 Action을 개발할지, 새로운 오브젝트를 생성할지 등에 대한 결정을 할 수 있다.

개발 및 운영 단계에서 오브젝트에 변동이 생겼을 경우 즉시 업데이트를 해주어 최신성을 유지해 주는 것 또한 중요하다.



BOL Template

BUSINESS OBJECT LIBRARY

Application Type (Technology)	Business Object Name	Actions	Inputs	Outputs	
Application Type 1	APP 1 VBO 1	Launch			
		Attach		Attached	
		Exit			
	APP 1 VBO 2	Attach			Attached
		Login	Username Password		
		Log out			
Get Logged in Status				Logged In Status	
Application Type 2	APP 2 VBO 1	Attach		Attached	
		Select Menu			
		Select Option	Option Name		
Application Type 3	APP 3 VBO 1	Launch			
		Attach		Attached	
		Exit			
	APP 3 VBO 2	Attach			Attached
		Login	Username Password		
		Log out			
Application Type 4	APP 4 VBO 1	Get Logged in Status		Logged In Status	
		Attach		Attached	
		Select Menu			
		Select Option	Option Name		



Business
Library

Business Object
Library

Business Object
Library

Business Object
Library



비즈니스 오브젝트 Best Practice

5가지 황금 룰



다중 오브젝트 디자인

동시에 여러 개발자가 오브젝트 개발 가능, 메모리 및 데이터베이스 효율성 증가, 오브젝트의 수정은 적은 프로세스에 영향 등의 이점.



단일 기능으로 제한

단일 기능만 수행하도록 개발된 Action은 보다 직관적이며 코드 수정 시 최소한의 영향도를 가져, 유지보수에 용이.



비즈니스 로직 제거

특정 업무에 국한된 비즈니스 데이터 또는 로직을 제거함으로써 오브젝트는 일반적인 설계가 되며, 결과적으로 재사용 성이 높아짐.



입력 매개변수 사용

Action 수행에 필요한 비즈니스 데이터 또는 로직을 입력 매개변수를 통해 Action에 전달하도록 설계.



출력 매개변수 사용

Action 수행의 결과는 출력 매개변수를 통해 호출자에게 반환. 프로세스는 해당 값에 따라 업무의 흐름을 결정.



블루프리즘 비즈니스 오브젝트



재사용성

어떤 자동화 프로세스 에도 구매 받지 않는, 대상 어플리케이션 또는 시스템과 상호작용하는 독립적인 오브젝트를 생성함에 따라 비즈니스 로직과 분리되어 높은 재사용성을 보장함.



확장성

재사용성이 높은 비즈니스 오브젝트들이 축적됨에 따라 새로운 자동화 프로세스를 제작하기 용이하며, 궁극적으로, 사내 코어 시스템들이 모두 오브젝트화 되었을 때 프로세스 정의만으로 손쉽게 확장할 수 있음.



효율성

객체지향 개념에 충실한 비즈니스 오브젝트 설계로 대상 어플리케이션의 변화에 대응하기 위해 프로세스를 수정할 필요 없이 오브젝트의 수정만으로 효율적으로 대응 가능.



유지보수 용이성

단일, 단순한 기능으로 제한된 Action들은 디버깅/수정 시 고려해야 할 Side-effects의 예상이 쉽고 직관적이기 때문에, 개발 공수가 줄어들고 안정성 또한 쉽게 확보할 수 있음.



표준화

기업의 거버넌스를 따르며 일관성 있는 설계로 구현된 오브젝트들은 기업의 자동화 시스템을 안정적이고 지속적으로 운영 및 확신을 할 수 있도록 표준화를 하는데 큰 기여를 함.

What's

Blueprism Processes



블루프리즘 프로세스란?

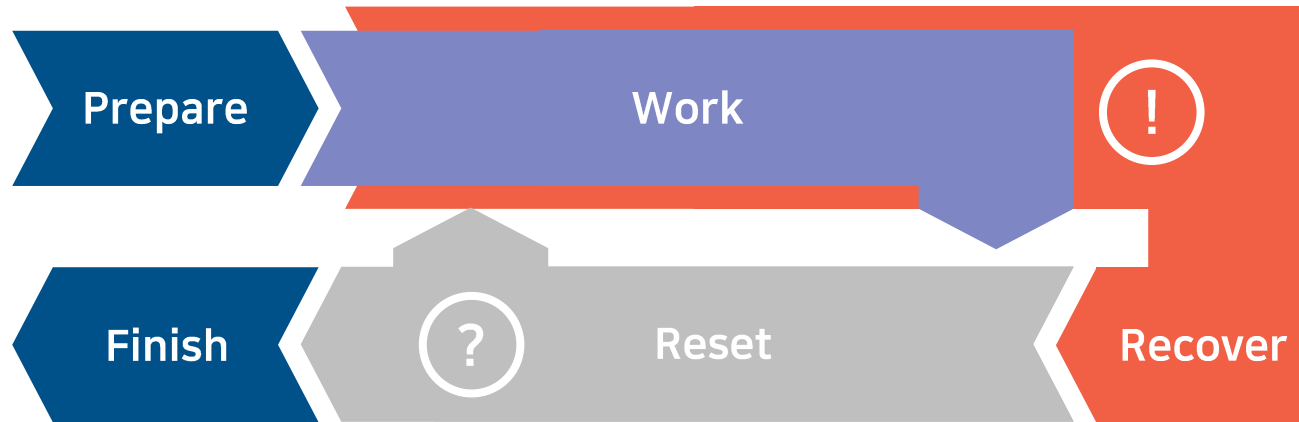
블루프리즘의 자동화 프로세스는 업무의 흐름을 쉽게 파악할 수 있는 워크플로우 기반으로 설계된다.

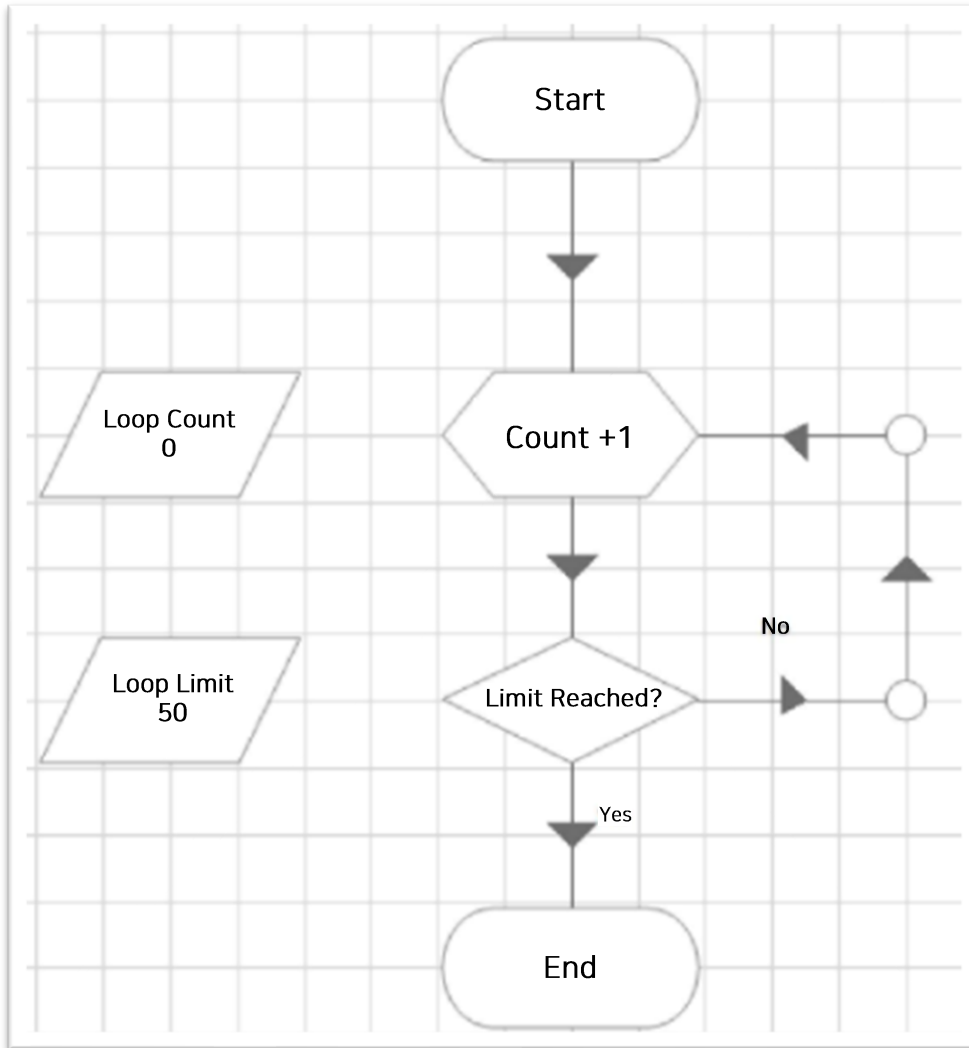
프로세스는 대상 어플리케이션과 상호작용을 담당하는 비즈니스 오브젝트에 정의된 Actions를 호출하여 업무의 흐름을 정의한다. 비즈니스 또는 시스템 예외가 발생했을 경우 어떻게 처리하는지에 대한 정의도 필요하다.

프로세스의 핵심은 예외 발생 시 이를 처리할 수 있어야 하고 예상치 못한 예외가 발생하더라도 강제종료가 되는 않으며 탄력적으로 복귀하여 기존의 작업을 이어갈 수 있도록 설계해야 한다.

블루프리즘은 신뢰성 높고 탄력적인 프로세스를 구성할 수 있는 프로세스를 만들기 위한 '예외처리 기법', '프로세스 템플릿', '워크 큐' 등과 같은 기능들을 제공한다.

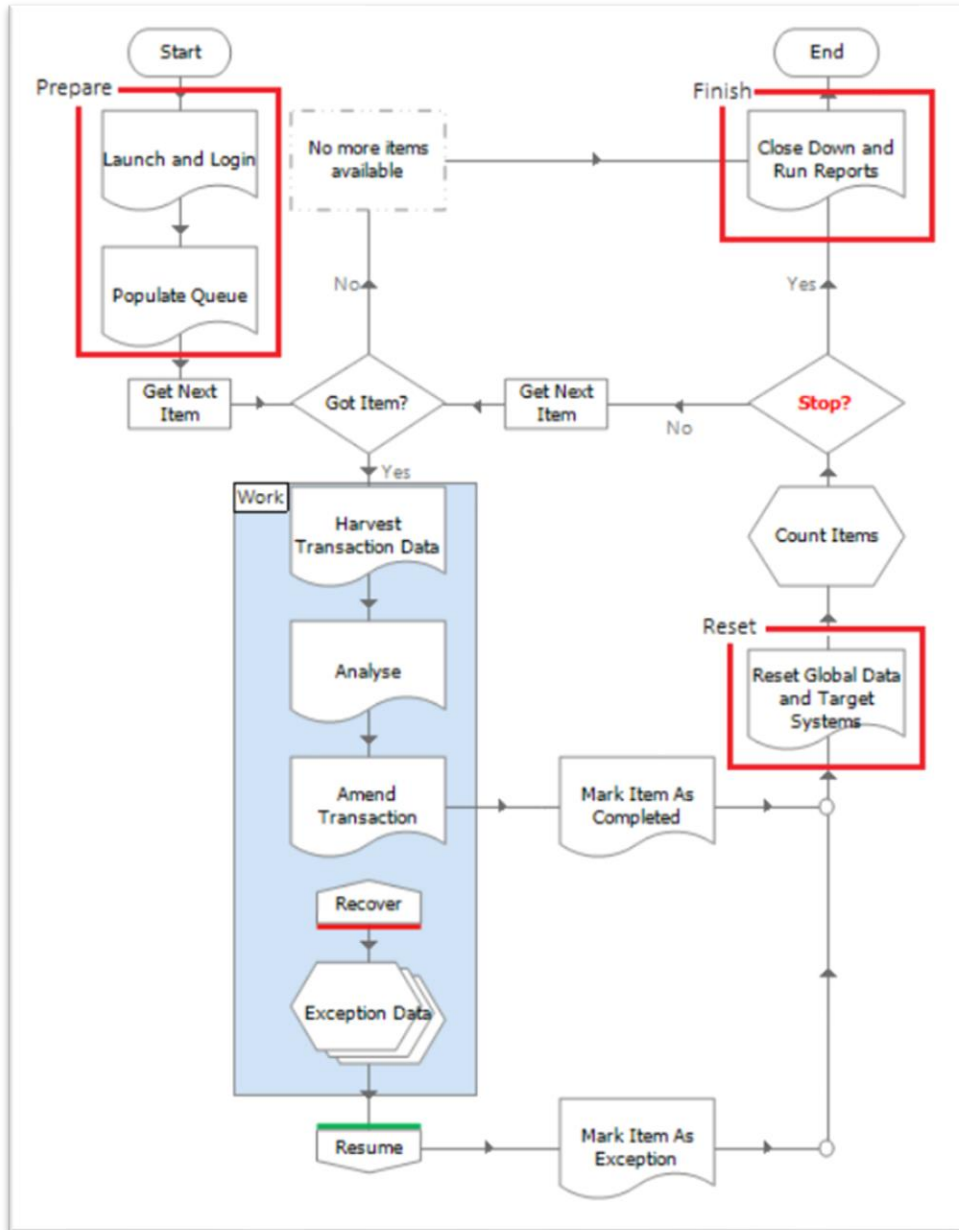
블루프리즘 프로세스의 컨셉





Circular Path

다량의 트랜잭션을 처리하는 프로세스의 기본은 반복 작업.
 블루프리즘의 프로세스의 골격이 어떻게 구체화 되는지 알아보기
 전에 가장 기본적인 순환 경로의 이미지를 기억한다.



Process Template

새 프로세스를 시작하기 위한 기본 프레임 워크.

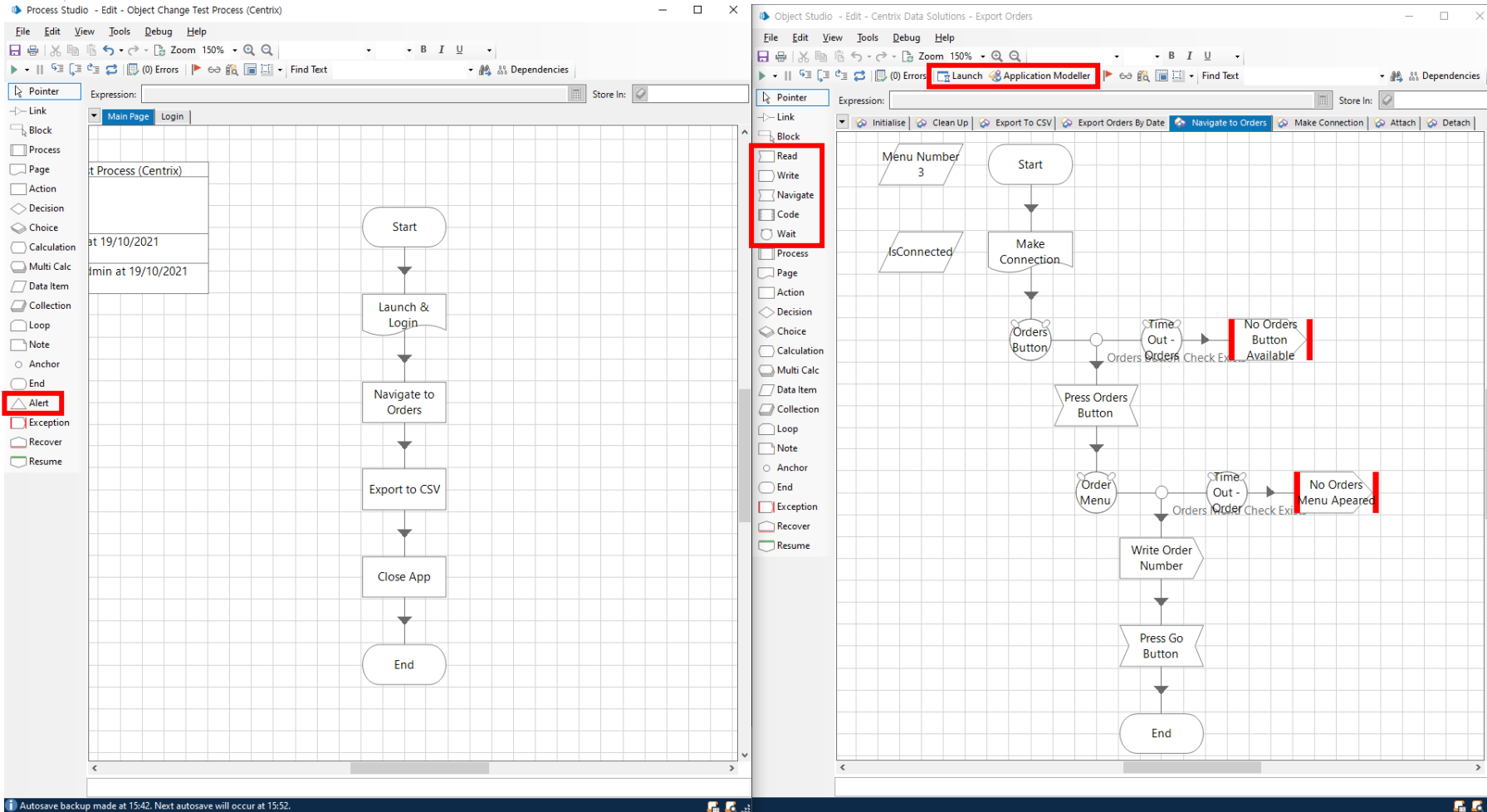
- 프로세스 개발 시간 단축
- 모든 프로세스에 일관성을 제공하여 업무의 흐름 파악 및 유지보수가 용이
- 개발자가 쉽게 블루프리즘 프로세스의 Best-practice와 조직의 개발 방법론을 준수

[Download Template](#)

* 로그인 필요



프로세스 스튜디오 vs 오브젝트 스튜디오

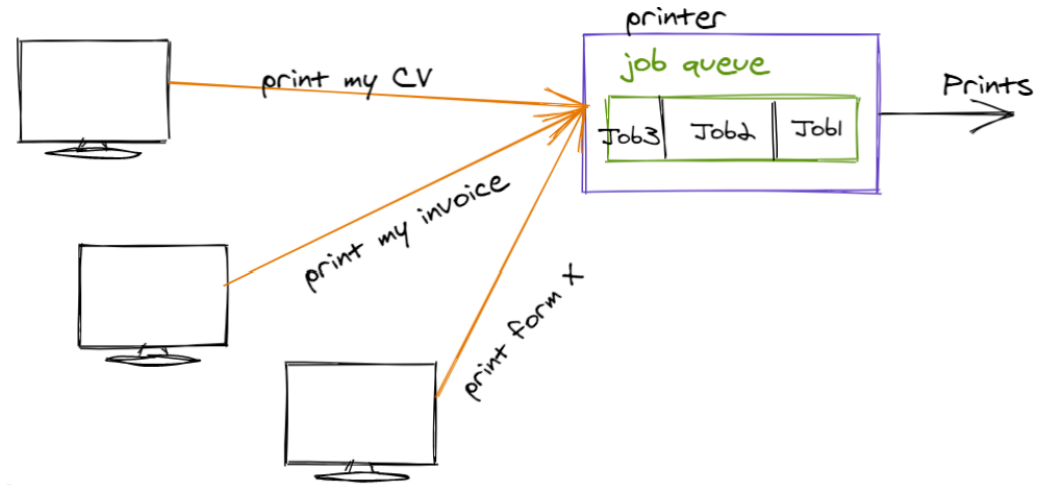


- Process Studio
 - Alert

- Business Studio
 - Read
 - Write
 - Navigate
 - Code
 - Wait
 - Application Modeller

What's

Work Queue



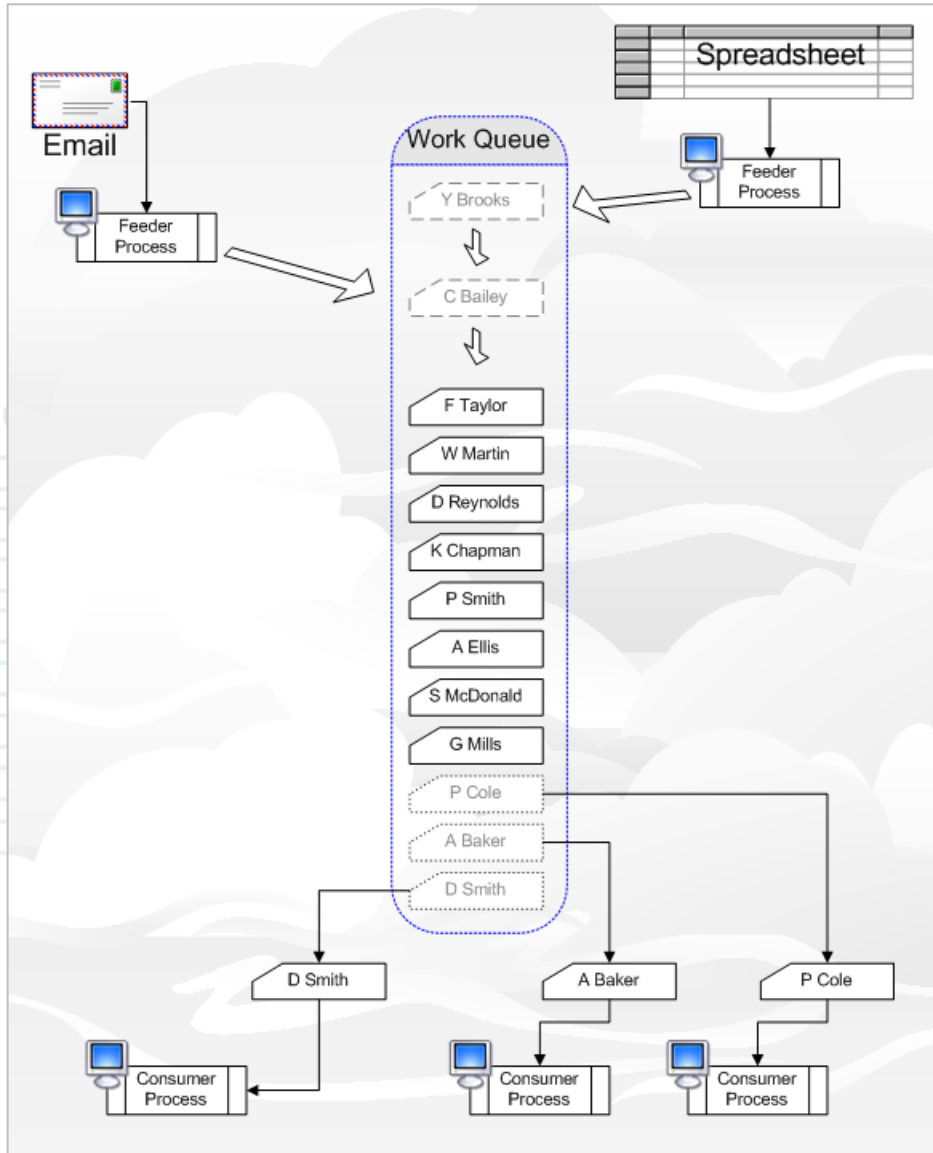
Work Queues of Blue Prism

블루프리즘은 기본적으로 모든 작업은 워크큐에 적재하여 수행하는 것을 Best Practice로 제시한다.

워크큐의 기능은 작업을 적재하고, 여러 런타임 리소스에서 효율적으로 병렬 처리를 가능케 해주며, 수행 결과를 기록하여 작업을 빠짐없이 추적하고 관리할 수 있는 기능을 제공한다. 이러한 기록을 한 눈에 볼 수 있는 워크큐 보고서 또한 제공한다.

운영자는 워크큐에 적재된 작업 리스트를 관리하며 작업량이 밀린 워크큐에 추가 런타임 리소스를 할당할지, 실패한 작업들을 재시도하거나 비즈니스 유저에게 수작업을 요청할지 등의 의사결정을 하고 이를 수행할 수 있다.

다시 말해, 워크큐는 자동화 프로세스들이 핸들링 하는 모든 작업을 한 눈에 모니터링하고 컨트롤 할 수 있는 기능이다.



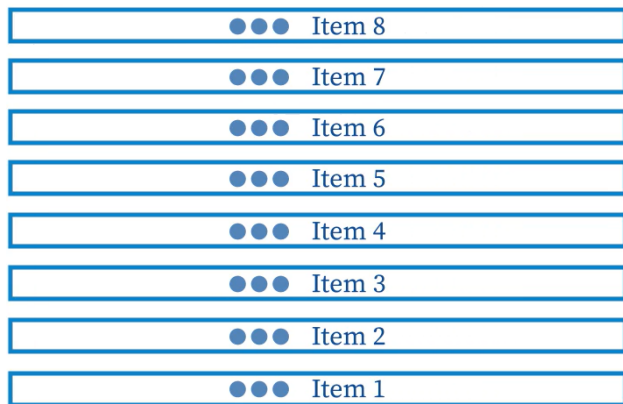
Introduction to Work Queue

블루프리즘 프로세스는 **반복적인 작업을 수행하도록 설계**되었습니다. 그리고 프로세스가 작업해야 하는 작업 아이템은 **목록 형식**일 수 있는 외부 소스에서 제공합니다.

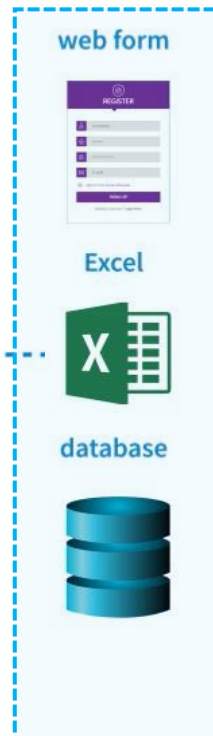
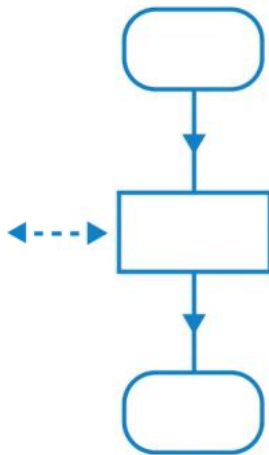
실제로 목록에 있는 대부분의 Item은 문제 없이 처리되지만 일부 아이템은 문제가 발생할 수 있습니다. 아이템이 'Completed'이든 'Exception'으로 Flag가 지정되면 강력한 감사를 위해 처리 결과를 기록해야 합니다.

목록의 소스에서 직접 Work하는 것은 철저한 모니터링 수준을 가능하게 하지 않습니다. 그렇기 때문에 Blue Prism Work Queue를 사용하면 처리 결과를 기록하고 저장, 관리, 공유 및 보고하는 기능이 내장된 Queue로 프로세스가 Item을 컴파일할 수 있으므로 항상 Blue Prism Work Queue를 사용하는 것이 가장 좋습니다.

Work Queue



Process

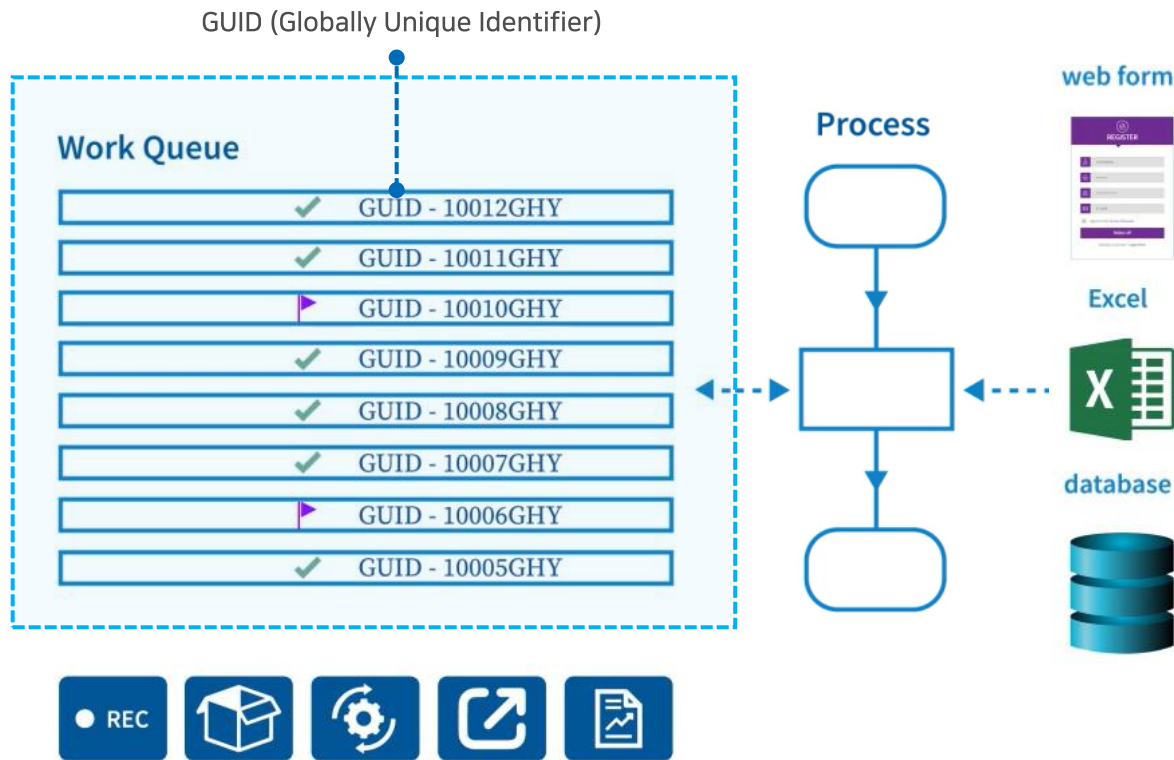


Introduction to Work Queue

블루프리즘 프로세스는 **반복적인 작업을 수행하도록 설계**되었습니다. 그리고 프로세스가 작업해야 하는 작업 아이템은 **목록 형식**일 수 있는 외부 소스에서 제공합니다.

실제로 목록에 있는 대부분의 Item은 문제 없이 처리되지만 일부 아이템은 문제가 발생할 수 있습니다. 아이템이 'Completed'이든 'Exception'으로 Flag가 지정되면 강력한 감사를 위해 처리 결과를 기록해야 합니다.

목록의 소스에서 직접 Work하는 것은 철저한 모니터링 수준을 가능하게 하지 않습니다. 그렇기 때문에 Blue Prism Work Queue를 사용하면 처리 결과를 기록하고 저장, 관리, 공유 및 보고하는 기능이 내장된 Queue로 프로세스가 Item을 컴파일할 수 있으므로 항상 Blue Prism Work Queue를 사용하는 것이 가장 좋습니다.



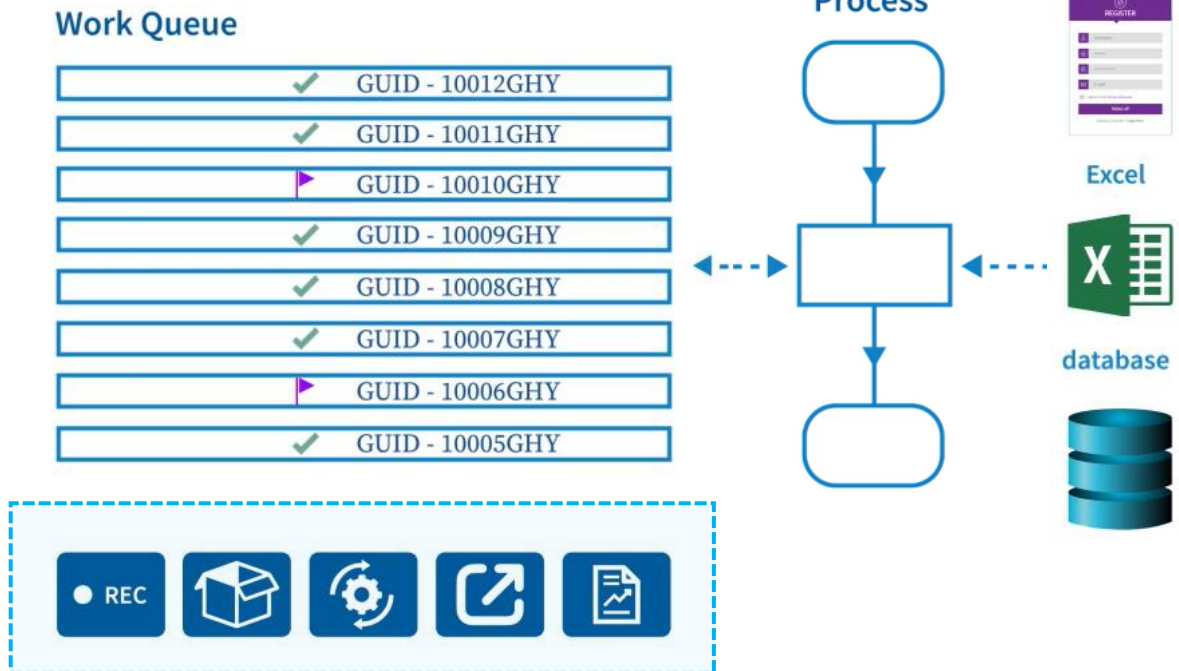
Introduction to Work Queue

블루프리즘 프로세스는 반복적인 작업을 수행하도록 설계되었습니다. 그리고 프로세스가 작업해야 하는 작업 아이템은 목록 형식일 수 있는 외부 소스에서 제공합니다.

실제로 목록에 있는 대부분의 아이템은 문제 없이 처리되지만 일부 아이템은 문제가 발생할 수 있습니다.

아이템이 'Completed'이든 'Exception'으로 Flag가 지정되든 강력한 감사를 위해 처리 결과를 기록해야 합니다.

목록의 소스에서 직접 Work하는 것은 철저한 모니터링 수준을 가능하게 하지 않습니다. 그렇기 때문에 Blue Prism Work Queue를 사용하면 처리 결과를 기록하고 저장, 관리, 공유 및 보고하는 기능이 내장된 Queue로 프로세스가 Item을 컴파일할 수 있으므로 항상 Blue Prism Work Queue을 사용하는 것이 가장 좋습니다.

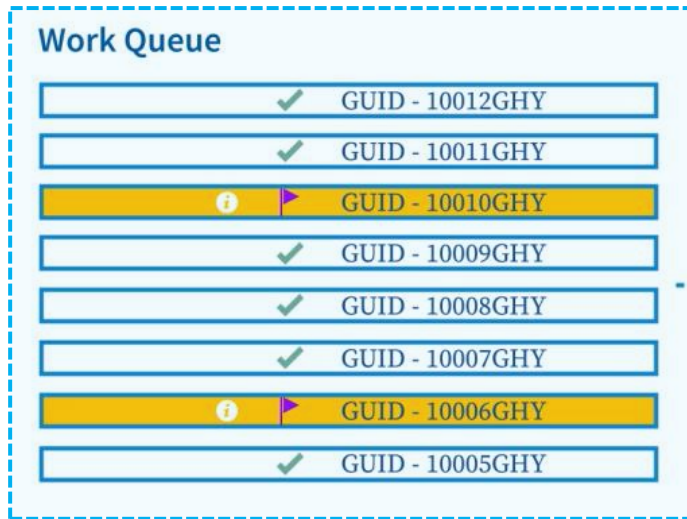


Introduction to Work Queue

블루프리즘 프로세스는 반복적인 작업을 수행하도록 설계되었습니다. 그리고 프로세스가 작업해야 하는 작업 아이템은 목록 형식일 수 있는 외부 소스에서 제공합니다.

실제로 목록에 있는 대부분의 Item은 문제 없이 처리되지만 일부 아이템은 문제가 발생할 수 있습니다. 아이템이 'Completed'이든 'Exception'으로 Flag가 지정되면 강력한 감사를 위해 처리 결과를 기록해야 합니다.

목록의 소스에서 직접 하는 것은 철저한 모니터링 수준을 가능하게 하지 않습니다. 그렇기 때문에 블루프리즘 워크큐를 사용하면 처리 결과를 기록하고 저장, 관리, 공유 및 보고하는 기능이 내장된 큐로 프로세스가 아이템을 컴파일할 수 있으므로 항상 블루프리즘 워크큐를 사용하는 것이 가장 좋습니다.



Process



web form



Excel



database

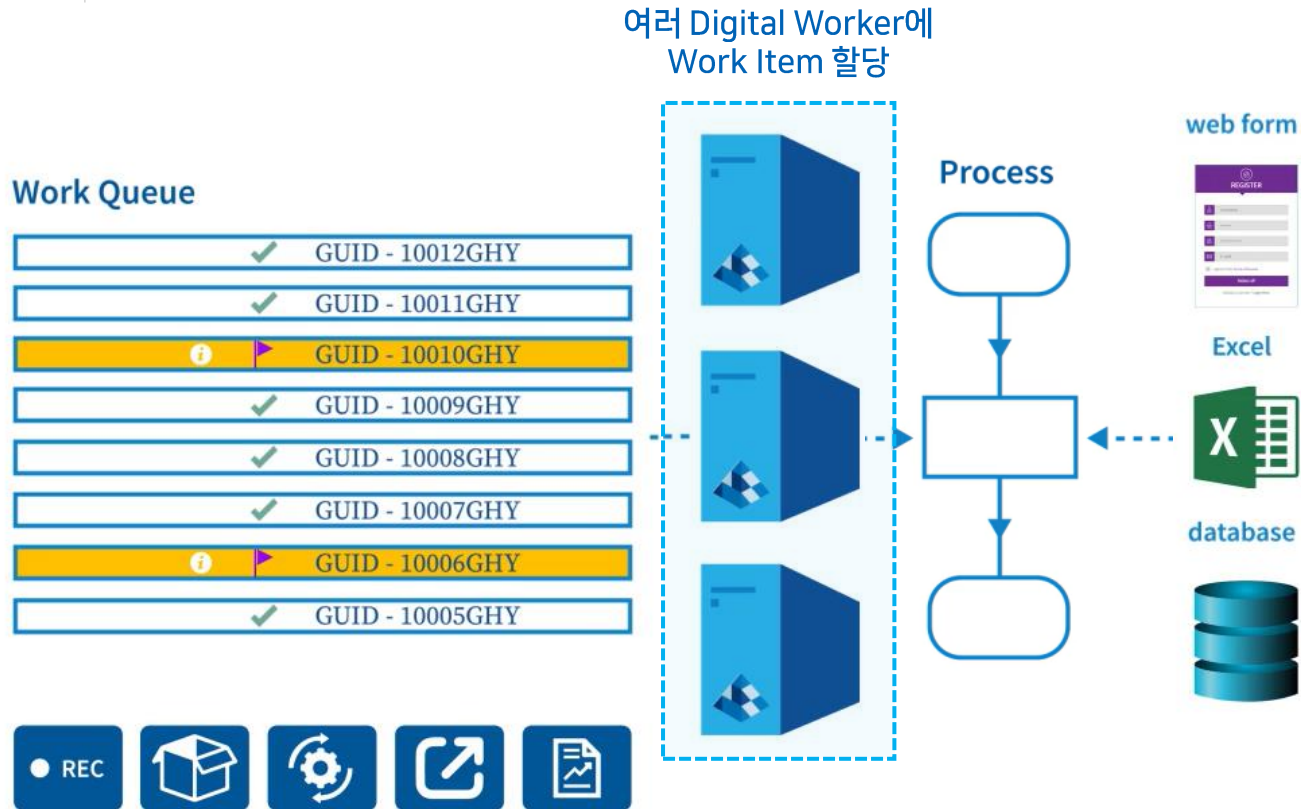


Introduction to Work Queue

워크큐는 기본적으로 채워진 다음 프로세스에서 작업 아이템의 목록입니다. 이 목록은 프로세스가 자체 Workload를 관리할 수 있도록 구성할 수 있습니다.

큐 내에서 아이템을 작업하는 동안 프로세스에 문제가 발생하면 해당 아이템에 문제에 대한 정보가 Tag로 지정될 수 있으며 이 정보는 수동 검토를 지원할 준비가 되어 워크큐에 저장됩니다.

하나의 블루프리즘 프로세스가 할당된 시간에 처리하기에는 아이템이 너무 많은 경우 여러 디지털 워커에서 동일한 프로세스를 실행하여 작업량을 공유할 수 있습니다. 여러 디지털 워커가 동일한 워크큐에서 동시에 아이템을 요청할 수 있으며, 각각은 처리할 워크큐에서 서로 다른 아이템을 수신합니다.



Introduction to Work Queue

워크큐는 기본적으로 채워진 다음 프로세스에서 작업 아이템의 목록입니다. 이 목록은 프로세스가 자체 Workload를 관리할 수 있도록 구성할 수 있습니다.

큐 내에서 아이템을 작업하는 동안 프로세스에 문제가 발생하면 해당 아이템에 문제에 대한 정보가 Tag로 지정될 수 있으며 이 정보는 수동 검토를 지원할 준비가 되어 워크큐에 저장됩니다.

하나의 블루프리즘 프로세스가 할당된 시간에 처리하기에는 아이템이 너무 많은 경우 여러 디지털 워커에서 동일한 프로세스를 실행하여 작업량을 공유할 수 있습니다. 여러 디지털 워커가 동일한 워크큐에서 동시에 아이템을 요청할 수 있으며, 각각은 처리할 워크큐에서 서로 다른 아이템을 수신합니다.

Working Items – Work Queues (Business Object)

블루프리즘은 소프트웨어에 사전 설치된 일부 비즈니스 오브젝트와 함께 제공됩니다.

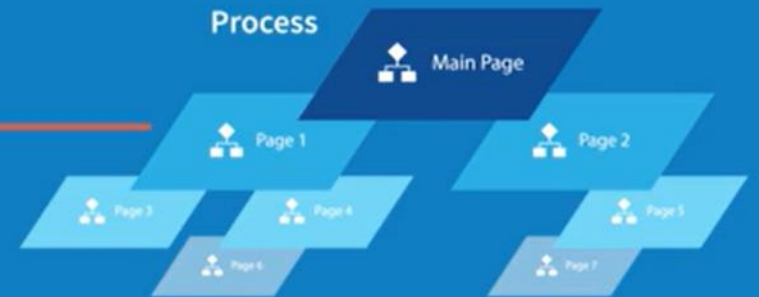
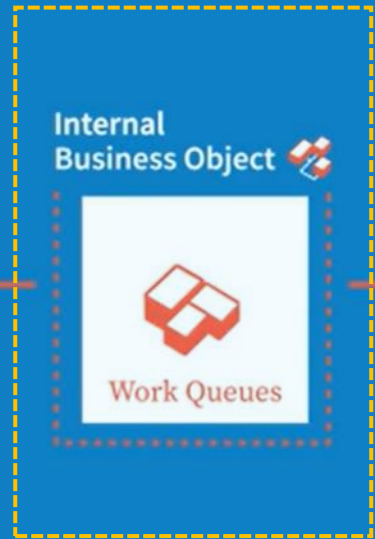
이러한 'Internal Business Object'는 다이어그램이 아니며 보거나 변경할 수 없으며 Action Stage를 통한 프로세스에서만 사용됩니다.

이것은 'Internal Business Objects - Work Queues Business Object' 로, 프로세스와 워크큐간의 연결을 제공하는 역할을 합니다.

외부 소스에서 추출한 데이터를 워크큐 내에 저장하는것을 시작으로 프로세스 를 시작할 수 있습니다.

Work Queue

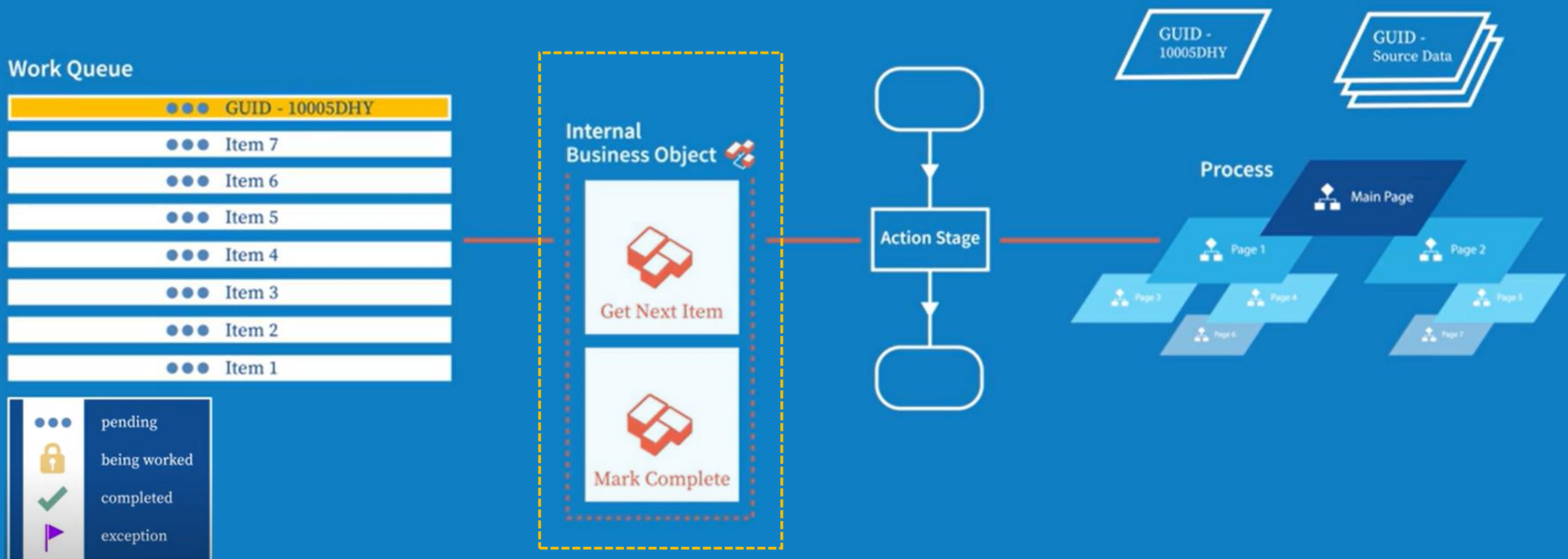
●●●	Item 8
●●●	Item 7
●●●	Item 6
●●●	Item 5
●●●	Item 4
●●●	Item 3
●●●	Item 2
●●●	Item 1



프로세스와 워크큐간
연결을 제공

Working Items – Get Next Item Action / GUID

Work Queues 비즈니스 오브젝트 내의 'Get Next Item' 은 프로세스에서 작업할 Queue 의 첫 번째 Pending Item을 수집합니다. Item이 Work Queue에 추가되면 'Globally Unique Identifier' 또는 'GUID'라는 고유 ID가 할당됩니다. 이 ID는 일련의 문자와 숫자로 구성되며 큐 아이템이 프로세스로 가져올 때 프로세스 Data Item 내에 저장됩니다. 아이템과 관련된 소스 데이터는 수집 단계에 저장됩니다.



Working Items – State

Locked : 작업을 위해 선택되어 아직 작업 중인 아이템의 상태 (잠긴 아이템은 둘 이상의 세션에 의해 큐에서 두번 선택되는 것을 방지함)

Exception : 해당 아이템이 작업을 위해 선택되었지만 세션에서 오류가 발생하거나 케이스가 작업을 수행할 프로세스에 적합하지 않아 완료되지 않은 상태

Pending : 아직 작업할 예정인 아이템의 상태, 해당 아이템은 지정된 순서대로 다음 Get Next Item 액션에 의해 선택됨

Completed : 아이템의 작업이 성공적으로 완료되어 더 이상 주의가 필요하지 않은 상태

Work Queue

- GUID - 10005DHY
- Item 7
- Item 6
- Item 5
- Item 4
- Item 3
- Item 2
- Item 1

IFA Applications	Running	0
New Invoices	Running	2
Pending Reconciliations	Running	0
Processed Invoices	Running	0

Queue Contents [Clear Filters](#) [Show Positions in Queue](#)

Item Key	Priority	Status	Tags
Being Worked, Locked(잠김)	0	Source	Source
Exception(예외)	0	Defer: I	Source
Pending(보류)	0	Source	Source
Completed(완료)	0	Source	Defer: I
	0	Source	Source

- pending
- being worked
- completed
- exception

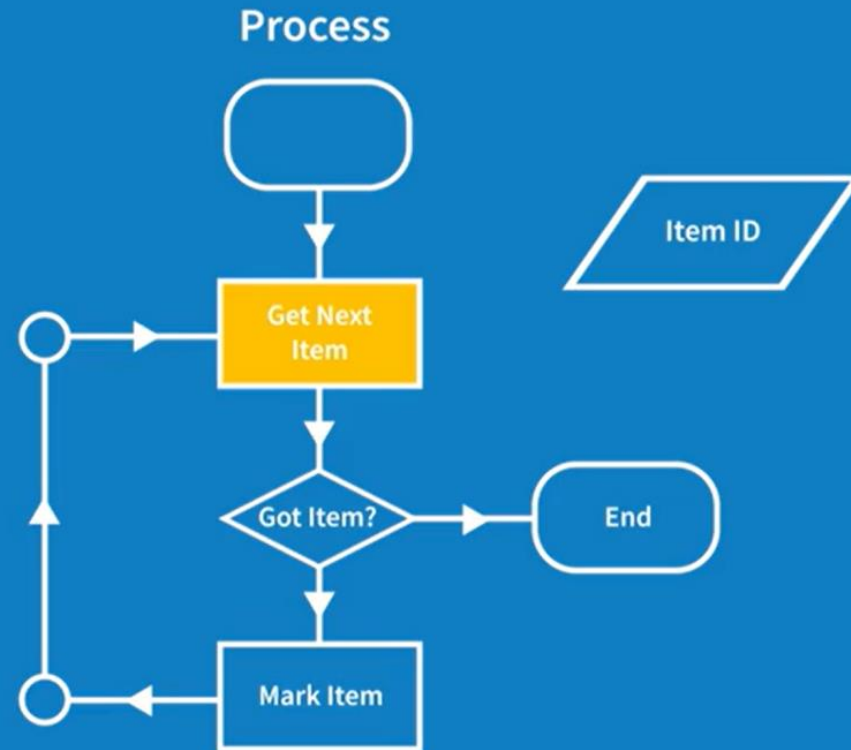


Queue Items – Circular Path

프로세스가 큐 내의 모든 아이템을 작업할 수 있도록 하려면 'Get Next Item' Action Stage로 흐름을 다시 반환하는 순환 경로가 프로세스에 포함되어야 합니다. 모든 아이템이 작업되고 'Completed' 또는 'Exception'로 표시되고 'Get Next Item' Action Stage' 가 빈 'Item ID'를 반환하면 프로세스가 종료 단계로 진행됩니다. 비어 있는 'Item ID' 는 작업으로 인해 워크큐에 더 이상 보류 중인 아이템이 없음을 나타냅니다.

Work Queue

✓	GUID - 10012GHY
✓	GUID - 10011GHY
▶	GUID - 10010GHY
✓	GUID - 10009GHY
✓	GUID - 10008GHY
✓	GUID - 10007GHY
▶	GUID - 10006GHY
✓	GUID - 10005GHY



Item Volumes	Sun 01	Mon 02	Tue 03	Total	Rate
Loaded	1				-
Worked		2			-
Completed			3		4
Business Referral	5			6	7
System Exception		8			
System Exception Rate			9	-	-

Median Case Times	Sun 01	Mon 02	Tue 03	Overall
Worked	10			-
Completed		11		-
Business Referral			12	-
System Exception				13

Attempts	Sun 01	Mon 02	Tue 03	Total
Completed After Multiple Attempts	14			-
Exception Despite Multiple Attempts		15		16

Business Referral Types	Sun 01	Mon 02	Tue 03	Total	Rate
17	18			19	20

System Exception Types	Sun 01	Mon 02	Tue 03	Total	Rate
21	22			23	24

Business Referrals	Sun 01	Mon 02	Tue 03	Total	Rate
25	26			27	28

System Exceptions	Sun 01	Mon 02	Tue 03	Total	Rate
29	30			31	32

Tags	Sun 01	Mon 02	Tue 03	Total
33	34			35

Queue Reporting - Performance Report

Performance Report는 지정된 기간에 대한 큐 결과의 개요를 생성합니다. 개별 아이템을 나열하는 대신 일일 큐 아이템들의 합계, 추세, 평균 및 완료율을 표시합니다.

Queue Report와 달리 Performance Report 템플릿은 고정되어 있으므로 사용자가 정의할 수 없습니다. 고급 사용자는 로고 또는 색상표를 변경할 수 있지만 템플릿 구조를 변경하면 손상될 위험이 있으므로 권장하지 않습니다.

How to read Performance Report

1. Items added to the queue on Sunday 1st.
 2. Items worked (completed or exception) on Monday 2nd.
 3. Items completed on Tuesday 3rd.
 4. Percentage of items worked during the period that were completed.
-



RPA 표준화

1

객체지향 오브젝트

블루프리즘에의 Best Practice를 따르는 재사용성이 높은 오브젝트를 통해 불필요한 비용 증가를 최소화 한다.

2

일관된 프로세스

블루프리즘의 프로세스 템플릿 및 워크큐를 활용하여 체계적이고 일관성 있는 프로세스를 제작한다.

3

유지보수의 용이성

객체지향 오브젝트와 일관된 프로세스로 인해 유지보수가 보다 용이하고 비용을 최소화 한다.

* RPA 확산 전, 반드시 이루어야 할 것은 유지보수의 안정화이다.

4

확산성

품질 높은 오브젝트로 자산화 된 기업의 시스템/어플리케이션들은 프로세스 설계 및 구현만으로 빠르고 비용효율적으로 확산 된다.

RPA 도입 후 3년

I RPA 여정에서의 어려움

RPA가 확산 되지 못하는 이유는 향후의 운영과 유지보수에 대한 준비가 없었고, 표준화와 숨은 비용에 대한 검토가 없었음

운영/유지보수의 어려움

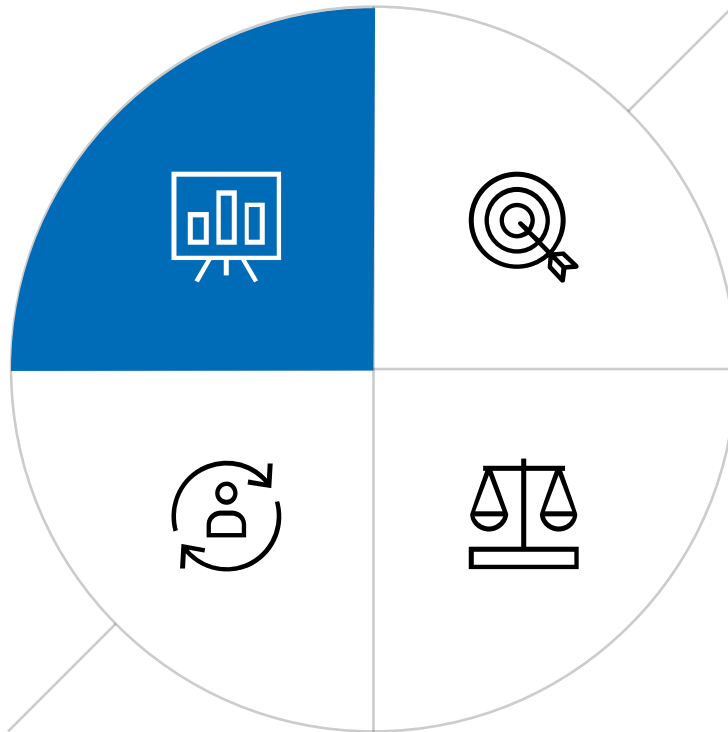
수시로 변경과 수정이 필요하다는 생각 X
표준화를 했어야 했다는 생각 X

개발자가 떠나고 나면 그 로직을 파악하기 어려움
인수인계나 변경/수정이 어려움
오류나 에러의 원인을 파악하기 어려움

경영진 설득의 어려움

실무진 선에서 추진하면 회사 경쟁력 제고될 거라는 착각
전사 프로세스 최적화보다는 손쉬운 Task 자동화에 집중

가치 창출의 한계로 추가 예산 확보하기 어려움
5차, 6차 등으로 확산하기 어려움



투자-대비-효과 (ROI)에 대한 의문

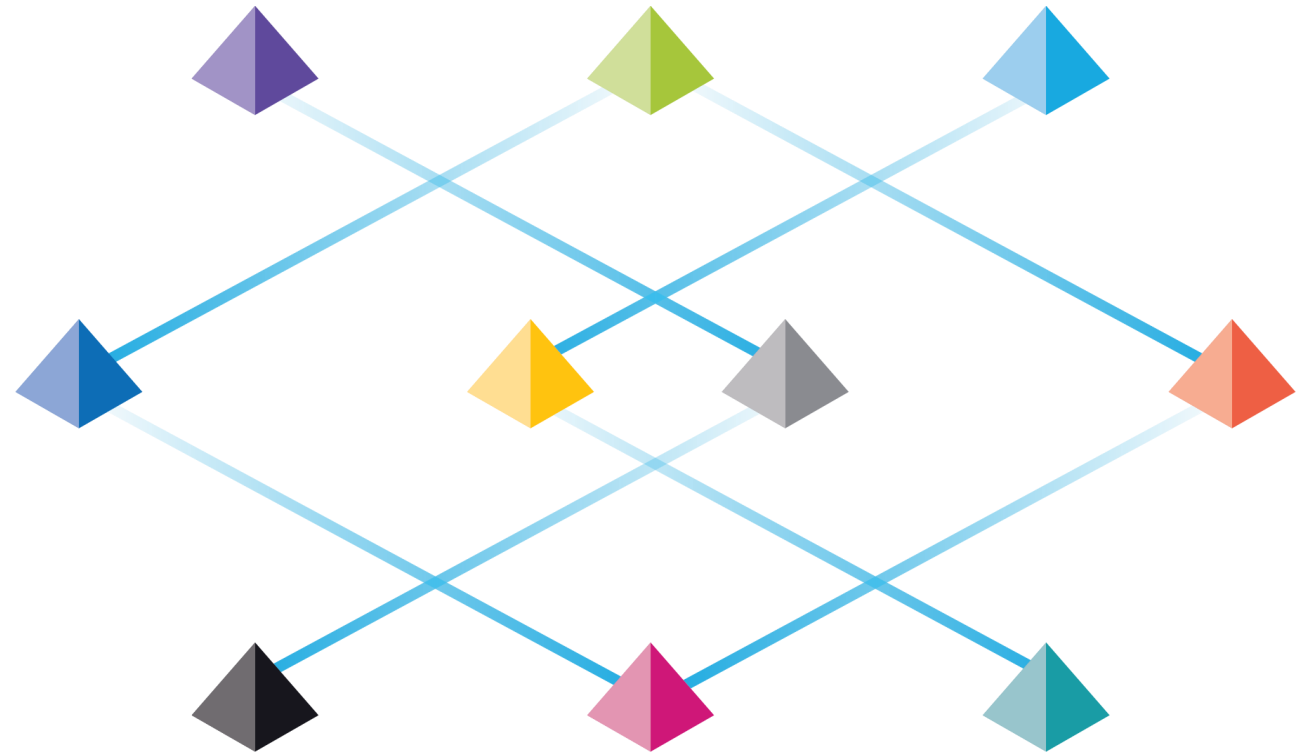
미래의 운영 비용에 대한 생각 X
확산 시에 라이선스에 숨은 비용에 대한 검토 X

운영비용이 개발비용 못지 않게 발생하기 시작
시민개발자에 대한 한계를 실감
내부 인력에 대한 ROI 압박이 시작

관리와 통제의 부재 -> 신뢰성 하락

로봇의 관리와 통제에 대한 생각 X
모니터링 해 주는 것을 관리와 통제가 되는 것으로 착각

로봇이 하는 일에 대한 신뢰성 결여
(신뢰성 = 안정성 + 보안성 + 감사성 + 추적성)
형상관리 (버전 컨트롤)는 거의 불가능



Thank you