ADM Unit 3 | Mission Brief 3.2

Offline Programming Deployment

Develop and debug a complete robot task in simulation before its first physical run.

Customer	Department	Difficulty	Time
Orbital Components Group	Virtual Commissioning	Advanced	2-3 class periods

Production Goal: Deploy a new robot task with minimal physical downtime.

Background

A spacecraft-component line cannot pause while engineers experiment with new code. Your team must program a replacement handling task offline, verify it virtually, and deploy it with minimal physical-cell downtime.

Learning Objectives

- Develop robot logic in an offline environment
- Use virtual testing to identify collisions and errors
- Create a controlled deployment checklist
- Compare predicted and actual commissioning time

Equipment

- Simulation/digital twin environment
- VEX CTE Workcell
- 6-Axis Arm
- Parts and fixtures
- Python project
- Deployment checklist

Student Instructions

1. Receive the required pickup, inspection, and placement sequence.
2. Measure or verify all fixture coordinates in the virtual model.
3. Write the full program without operating the physical robot.
4. Run at least ten simulated cycles and resolve every collision or fault.
5. Record predicted cycle time and create a risk checklist.
6. Perform a line-by-line safety review before physical deployment.
7. Run the physical system first at reduced speed.
8. Correct only differences that could not be predicted virtually.
9. Record total physical commissioning time and compare it with the predicted plan.

Engineering Requirements

- Complete ten fault-free simulated cycles before deployment
- Complete the first physical cycle at reduced speed
- Achieve five consecutive physical cycles
- Document every change made after deployment

Constraints & Safety

- No physical trial-and-error before simulation approval
- Hardware coordinates must be verified
- Safety stop must be tested before deployment
- Use approved reduced speed for first run

Deliverables

- Offline Python program
- Simulation test log
- Collision/risk checklist
- Deployment checklist
- Physical commissioning record
- Reflection

Reflection Questions

1. Which issue was discovered only in the physical cell?
2. How much physical downtime did offline programming save?
3. Which model detail was most important?
4. How should future fixtures be documented?

Engineering Support

Engineering Tip
Treat the first physical run as verification, not as the first debugging session.

Common Mistake
Assuming fixture coordinates are identical between virtual and physical cells can cause immediate collisions.

Stretch Goal
Deploy a second product variation using only parameter changes.

Career Connection
Offline robot programmers prepare and validate production code without interrupting active factory lines.

Before You Submit

- Notebook evidence included
- Requirements checked
- Testing documented
- Reflection completed